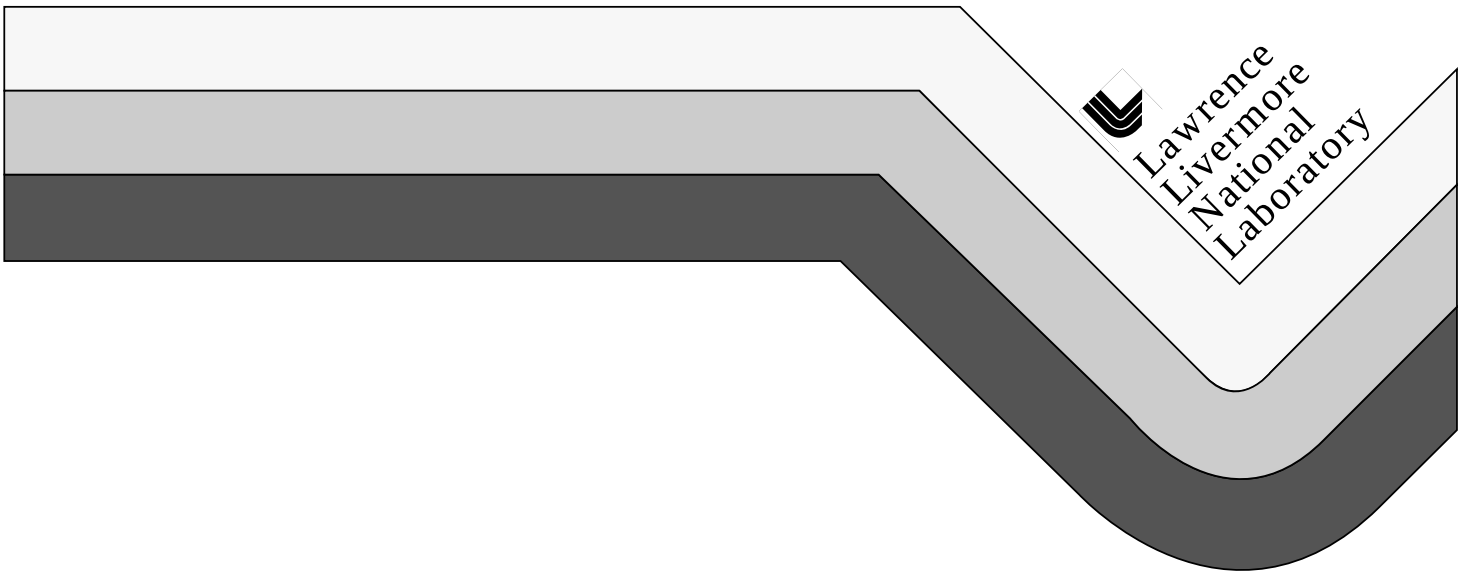
UCRL-MA-103581 Rev. 4

User's Guide to the CG-MATHEW/ADPIC Models
Version 4.1

Atmospheric Release Advisory Capability

November 1995



Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

UCRL-MA-103581 Rev. 4

USER'S GUIDE TO THE CG-MATHEW/ADPIC MODELS Version 4.1

Atmospheric Release Advisory Capability

November 1995

Atmospheric and Ecological Sciences Program
Lawrence Livermore National Laboratory
University of California
Livermore, California 94550

Acknowledgments

Because of the dynamic and evolutionary nature of the MATHEW/ADPIC models, and because their use needs to be well-documented for the Atmospheric Release Advisory Capability (ARAC) operations team, this manual has undergone numerous revisions of varying magnitude over the years. The table below indicates the authors (in alphabetic order) that contributed to the changes for this revision, as well as the accumulation of authors for the existing content of the manual:

Topic	Accumulation of Authors for Existing Material	Authors Contributing to Changes for Revision 4
Word processing & document preparation	Raylene Cooper, Jodi Greenfield, Doris Gresho	Jodi Greenfield
Editor	Connee Foster, Bryan Lawver, Dan Rodriguez	Connee Foster
Chapters 1 & 2 regarding TOPOG	Hoyt Walker	
Chapters 1 & 2 regarding MEDIC	Dan Rodriguez, Hoyt Walker	
Chapters 1 & 2 regarding MATHEW	Gayle Sugiyama	
Chapters 1 & 2 regarding ADPIC	Ron Baskett, Gordon Duckworth, Don Ermak, Connee Foster, Kevin Foster, Robert Freis, George Greenly, Rolf Lange, John Nasstrom, Brenda Pobanz, Allan Taylor	Don Ermak, Bob Freis, Brenda Pobanz, John Nasstrom, Allan Taylor
Chapters 1 & 2 regarding PLCNT & TIMEHIS	Allan Taylor, Hoyt Walker	
Chapters 1 & 2 regarding other info	Kevin Foster, Hoyt Walker	Hoyt Walker
Chapter 3	Ron Baskett, Gordon Duckworth, Don Ermak, Connee Foster, Kevin Foster, Robert Freis, George Greenly, Rolf Lange, Len Lawson, John Nasstrom, John Pace, Brenda Pobanz, Dan Rodriguez, Walt Schalk, Allan Taylor, Phil Vogt, Hoyt Walker	Gordon Duckworth, Connee Foster, Kevin Foster, Brenda Pobanz, John Nasstrom, Allan Taylor, Hoyt Walker
Appendix B	John Pace, Hoyt Walker	
Appendix C	Hoyt Walker	
Appendix D	Connee Foster, Gayle Sugiyama, Allan Taylor, Hoyt Walker	

Appendix E	Gordon Duckworth, Connee Foster, Brenda Pobanz, Gayle Sugiyama, Allan Taylor, Hoyt Walker	Connee Foster, John Nasstrom, Brenda Pobanz, Allan Taylor
Appendix F	Hoyt Walker	Hoyt Walker

This manual was originally released in 1982 with the following authors: Daniel J. Rodriguez, George D. Greenly, Philip M. Gresho, Rolf Lange, Bryan S. Lawver, Leonard A. Lawson, and Hoyt Walker, with Dan Rodriguez also acting as the editor. Connee Foster assumed the editor/co-author role for subsequent unofficial versions of the manual. Connee was also the editor for the next official version (Rev. 1) of the this document released in December 1992. She was ably assisted in preparing that version by: Robert P. Freis, Kevin T. Foster, John S. Nasstrom, Ronald L. Baskett, Walter W. Schalk III, James S. Ellis, Brenda M. Pobanz, Philip J. Vogt, Leonard A. Lawson, Hoyt Walker, and John C. Pace. Special thanks go to Raylene Cooper for her work in preparing and editing the December 1992 version.

A major re-write by Allan Taylor, Gayle Sugiyama, and Hoyt Walker occurred during Summer 1993, Rev. 2, reflecting the initial implementation of the restructured models. This restructuring was basically a general housecleaning of the structure of the models required to facilitate easier maintenance and easier insertion of upcoming major model improvements. Subsequent informal versions have been edited by Connee Foster with input provided primarily by Allan, Gayle, and Hoyt. Rev. 2 reflects the current state of the models after the restructuring and prior to inclusion of major improvements (see following paragraphs). Special thanks go to Jodi Greenfield for help with scanning figures and editing.

Version Rev 3 includes Conjugate Gradient MATHEW, generalized map projection capability, and some other minor changes and additions. Primary authors are Connee Foster, Hoyt Walker, Gordon Duckworth, Allan Taylor, and Gayle Sugiyama. Special thanks go to Jodi Greenfield for word processing and document preparation for this revision.

Version Rev 4 includes the Random Displacement Method (RDM) for calculation diffusion in ADPIC using a random walk, Monte Carlo approach, hybrid particles, moving receptors, plus other minor changes and additions. The right hand column of the table above shows the authors contributing to this revision.

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

Preface

This user's guide documents the CG-MATHEW and ADPIC transport and diffusion models and their supporting stream of codes. It includes a description of each code (Chapter I), its theoretical basis (Chapter II), and the preparation of input and output data (Chapter III).

Table of Contents

+ Abstracted from Sugiyama et al., 1994.

I. Introduction to the CG-MATHEW/ADPIC Transport and Diffusion Models

This user's guide provides both an operational and theoretical summary of the CG-MATHEW/ADPIC transport and diffusion models and supplemental codes needed to execute these models. The acronym CG-MATHEW comes from **C**onjugate-**G**radient **M**ass-**A**djusted **THrE**e-dimensional **W**ind field. ADPIC stands for **A**tmospheric **D**iffusion **P**article-**I**n-**C**ell model. These models calculate air concentration and ground deposition on a regional scale (up to 200 km) from continuous or instantaneous point sources.

CG-MATHEW is a three-dimensional meteorological data-adjustment model which supplies ADPIC with mean wind fields that are mass-consistent and representative of the available meteorological measurements (surface, tower, and upper air soundings). The bottom boundary in this model is determined by grid cell estimates of the actual topographic features of a given site and its environs, which can play a very important role in defining or modifying regional-scale flow patterns. ADPIC is a three-dimensional particle transport and diffusion code capable of calculating the time-dependent dispersion of inert or radioactive air pollutants.

In addition to CG-MATHEW/ADPIC, three supplemental codes are required to provide input/output support. These codes, which are also discussed in this user's guide, are TOPOG, MEDIC, PLCNT and TIMEHIS. TOPOG defines topographic features for MEDIC, CG-MATHEW, and ADPIC. MEDIC builds a realistic three-dimensional wind field from the observed meteorological conditions. This input is used by CG-MATHEW to compute the mass-consistent wind field required by ADPIC. PLCNT is a post-processing code that allows dose conversion from particle concentration to other units (e.g., Rem). PLCNT uses the resulting doses to generate products for shipment to ARAC site workstations.

An overview of CG-MATHEW/ADPIC and the support routines follows in this section. A complete technical formulation of each code is found in Section II. Section III provides a user's guide to each code.

I.A. CG-MATHEW

The Conjugate-Gradient Mass-Adjusted THrEe-dimensional Wind (CG-MATHEW) code generates mass-consistent winds by minimal adjustment of input fields derived from observational, analyzed, or model data. The adjustment is performed by a constrained variational minimization using finite-difference methods and a conjugate gradient solution. Generalized symmetric boundary conditions are implemented, which allow control of the relative amounts of normal and tangential adjustment at the lateral and terrain surfaces. Sugiyama et al. (1994) discuss the methodology in detail, give comparisons with potential flow solutions for simple geometries, and demonstrate the use of the new boundary conditions to create different kinds of flow fields, prevent speed reductions, and eliminate flow artifacts caused by complex terrain.

The underlying theoretical framework for the variational adjustment was developed by Sasaki (1958, 1970a, 1970b) and first applied in the Atmospheric Release Advisory Capability (ARAC) operational context by Sherman (1978). A functional is formulated which minimizes the variance between input and output winds subject to a non-divergence constraint, mass-flow boundary conditions on lateral and top faces, and the requirement of zero normal wind-field components at terrain surfaces. The requirement of minimal adjustment maintains consistency with available meteorological measurements (surface and upper air soundings), while atmospheric stability conditions govern the relative amounts of change in the vertical and horizontal wind components. The adjustment is irrotational and neither momentum balance nor energy conservation is imposed. Velocity components are defined on staggered grid faces so that the mass-consistency constraint is cell-flux rather than grid-point based. The problem then reduces to the solution of the Poisson equation in the Lagrange multiplier $\lambda(x, y, z)$, with the adjusted winds derived from the corresponding Euler-Lagrange equations.

The original ARAC MATHEW code discretized the Poisson equation using quadratic fits to generate the boundary conditions, which resulted in an asymmetric coefficient matrix. The equation was then solved by a successive overrelaxation (SOR) method. In order to implement a conjugate gradient algorithm, which is approximately twice as fast as the SOR approach, the problem has been reformulated with standard image point boundary conditions. This yields a symmetric matrix representation consistent with the symmetry of the continuous Poisson problem. The boundary conditions are generalized to include parameters for controlling the relative amounts of normal and tangential adjustment at the lateral and top edges (MIXED boundary condition, Gresho, 1978) and for fixing velocities at the inflow boundaries.

Input wind fields are generally derived by a combination of interpolation and extrapolation from sparse observational data and are typically assumed to be purely horizontal due to the lack of vertical velocity measurements. Such two-dimensional wind fields can be prepared for CG-MATHEW using the MEDIC code which incorporates surface, upper air, and analyzed wind data as available and appropriate. However, the mass-adjustment technique is applicable to input winds with a complete set of *u*, *v*, and *w* components and can be implemented using three-dimensional wind fields derived from assorted diagnostic and prognostic models.

I.B. ADPIC

ADPIC (Atmospheric Diffusion Particle-in-Cell) is a numerical, three-dimensional particle-diffusion code capable of calculating the time-dependent distribution of air pollutants under many conditions including strongly distorted wind fields, calm conditions, wet and dry deposition, radioactive decay, and space- and time-variable turbulence parameters.

ADPIC solves the three-dimensional advection-diffusion equation. This version of ADPIC includes a new option to use the Random Displacement Method (RDM) described by Ermak *et al*. (1995). The RDM is a random walk, Monte Carlo method of solving the advection-diffusion equation using Lagrangian marker particles to represent species mass. The original version of ADPIC solved the advection-diffusion equation using a Gradient Diffusion method that is a hybrid Eulerian-Lagrangian pseudo-velocity method developed by Lange (1973, 1978, and 1989), and described in earlier versions of this user's guide. This Gradient Diffusion method is retained as an option in this version of ADPIC. Mass-consistent mean wind fields supplied to ADPIC by CG-MATHEW.

The RDM diffusion calculation is a purely Lagrangian, grid-independent calculation. Some of the benefits of this approach compared to the previously used Gradient Diffusion method in ADPIC are (a) sub-grid diffusion approximation is no longer needed, (b) numerical accuracy of the diffusion calculation is improved because particle displacement does not depend on the resolution of the Eulerian grid used to calculate species concentration, and (c) future adaptation to other grid structures for the input wind field does not affect the diffusion calculation. In addition, the RDM incorporates a unique and accurate treatment of particle interaction with the surface.

I.C. CG-MATHEW and ADPIC Support Routines

I.C.1. TOPOG

TOPOG creates the underlying surface for the CG-MATHEW and ADPIC models. The surface is created by averaging terrain data from a regional terrain data base which has, as its source, high resolution elevation data supplied by the Defense Mapping Agency.

TOPOG has three main functions: (1) It verifies that the CG-MATHEW and ADPIC grids, as defined by the user, are oriented correctly with respect to each other and the regional terrain grid. (2) It averages terrain data from the regional data base to the newly defined model grid and then rounds these averaged heights to the nearest model level. This provides the appropriate block topography for the CG-MATHEW and ADPIC models. (3)

I.C.2. MEDIC

MEDIC extrapolates a three-dimensional gridded wind field from a combination of surface and upper air observations. The extrapolation is a complex process, controlled by a number of parameters specified by the user. The extrapolated wind field is sensitive to small changes in some of these parameters and, as a result, widely differing extrapolated fields can be produced, all of which are consistent with a given data set. CG-MATHEW adjusts the winds to achieve mass consistency and then passes them to ADPIC. Because CG-MATHEW makes the smallest necessary adjustments to the extrapolated field and because ADPIC assumes persistence over extended periods (typically one hour), any change in the extrapolated wind field will affect all the results of the ARAC models. Therefore, it is crucial for the user to provide the best extrapolated wind field that can be produced for a data set within the allowed range of the various parameter values. In turn, this means that the graphical output from MEDIC must be examined very carefully by the assessor to determine if the wind field is an acceptable model of the physical state of the atmosphere. If it is not, then MEDIC must be re-executed with revised parameter values until a reasonable representation is produced.

I.C.3. PLCNT

PLCNT uses as input two-dimensional gridded arrays of concentrations that are produced by ADPIC at user-selected heights in what are called "sampling bins". These 2-D sampling bin arrays are converted to "dose bin" arrays using a user-specified multiplicative "dose conversion factor", and then used individually or summed in to "final bin" arrays for subsequent processing. At user-defined times, PLCNT writes files with the final 2-D arrays of data for creating contour plots on the ARAC central system or transfer to ARAC site workstations. Up to five contours levels will selected by the code, each differing by an order of magnitude, as a function of the largest grid point concentration. The user has the option of overriding these contour level choices (see Section III.B.8). As an example of the usefulness of this latter option, an environmental standard may have been established for a

substance which has been accidentally released into the atmosphere. On entering this value, the code will plot a contour level, if present, which represents a threshold to this standard, thereby delineating the area of concern.

There are three types of model output data: total deposition, integrated air concentration, and instantaneous air concentration. Total deposition is cumulative mass or activity per unit area on the ground over the model domain since the beginning of the calculation. Integration air concentration is integrated in time over all horizontal grid points at selected heights. Instantaneous air concentration are calculated at specified times over all horizontal grid points at selected heights.

PLCNT also has a centerline concentration capability, which starts at a user-specified ADPIC source location and searches along arcs of increasing radius for the maximum along each radial arc. The result series of maxima, along with their locations are available in tabular format, and the locations are also plotted on each contour plot.

I.C.4. TIMEHIS

In addition to the ability to obtain contours from PLCNT, a companion code TIMEHIS (i.e., Time History) allows concentrations to be estimated using bilinear interpolation for point locations that can be arbitrarily positioned in the grid. The grid point concentrations are developed in exactly the same way as in PLCNT with interpolation to the final bins performed before any smoothing. The times at which the interpolations are made are also determined in the same way as the times of the plots in PLCNT. The series of concentrations are plotted as time series and printed in the TIMEHIS.LOG file. TIMEHIS will also read sampler measurement data, plotting these observations on the time series plots along with the computed values for the same locations. In addition, TIMEHIS generates statistics that compare the computed and measured values. These features have proved useful when comparing model concentrations against collections by actual samplers that have been placed in the field during a tracer experiment.

II. Technical Discussion of ARAC Models

II.A. CG-MATHEW+

II.A.1. Variational formulation

A variational problem may be solved by reduction to a set of differential equations known as the Euler equations, which represent necessary, though not sufficient, conditions for a minimum (Courant and Hilbert, 1962). In its simplest form, the CG-MATHEW functional required to minimize the variance between the observed input and adjusted output winds subject to the mass-consistency constraint is

$$J(u,v,w,\lambda) = \int_{\Omega} \frac{1}{2\sigma_H^2}\left(\left(u-u^o\right)^2+\left(v-v^o\right)^2\right) + \frac{1}{2\sigma_v^2}\left(w-w^o\right)^2 + \lambda\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial x}+\frac{\partial w}{\partial z}\right)dxdydz \qquad \text{(II.A.1)}$$

where (u, v, w) and $\left(u^o, v^o, w^o\right)$ are the output and input wind components, respectively, $\lambda(x, y, z)$ is the Lagrange multiplier for the mass-consistency constraint, Ω is the volume domain of the simulation, and σ_v and σ_H are parameters controlling the relative amounts of vertical and horizontal adjustment. The Euler-Lagrange equations associated with this functional are

$$\begin{aligned} u - u^o &= \sigma_H^2 \frac{\partial \lambda}{\partial x} \\ v - v^o &= \sigma_H^2 \frac{\partial \lambda}{\partial y} \\ w - w^o &= \sigma_V^2 \frac{\partial \lambda}{\partial z} \end{aligned} \qquad \text{(II.A.2)}$$

with boundary conditions

$$\begin{aligned} \lambda n_x \, \delta u\big|_s &= 0 \\ \lambda n_y \, \delta v\big|_s &= 0 \\ \lambda n_z \, \delta w\big|_s &= 0 \quad . \end{aligned} \qquad \text{(II.A.3)}$$

Taking appropriate derivatives of the Euler-Lagrange equations and substituting into the mass-consistency constraint, $\nabla \cdot \bar{u} = 0$, yields the Poisson equation

$$\sigma_H^2\left(\frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2}\right) + \sigma_V^2 \frac{\partial^2 \lambda}{\partial z^2} = -\frac{\partial u^0}{\partial x} - \frac{\partial v^0}{\partial y} - \frac{\partial w^0}{\partial z} \qquad \text{(II.A.4)}$$

which may be solved for λ subject to the boundary conditions. The adjusted wind fields are then determined from Eqs. II.A.2.

+ Abstracted from Sugiyama et al., 1994.

II.A.2. Boundary Conditions

The boundary conditions of Eqs. II.A.3 can be satisfied if $\lambda\big|_S = 0$. The physical interpretation of this solution is revealed by examination of the Euler-Lagrange equations. If λ is a constant on S, the derivatives along the surface vanish, which fixes the tangential component of the velocity, while imposing no restrictions on the normal component. This is the ADJUST_NORMAL condition and is a kind of flow-through boundary. Eqs. II.A.3 are also satisfied if $\frac{\partial\lambda}{\partial n}\big|_S = 0$, which prevents changes in the normal velocity component, while allowing free adjustment of the tangential component. This is a FIXED_NORMAL or fixed-flow boundary and can be used to maintain zero normal velocities on terrain surfaces [1].

It is advantageous to incorporate a new surface integral term which provides more flexibility in the boundary conditions and to treat the FIXED_NORMAL case in a constraint formalism, so that input wind fields with non-zero normal components at terrain surfaces can be used. These considerations lead to a generalized CG-MATHEW functional

$$\begin{aligned} J(u, v, w, \lambda) = & \int_\Omega \left[\left(\vec{u} - \vec{u}^0\right) \cdot \mathbf{D}^{-1} \cdot \left(\vec{u} - \vec{u}^0\right) + \lambda \nabla \cdot \vec{u} \right] dxdydz \\ & - \int_{S_{terrain}} \lambda_{terrain}\, \vec{n} \cdot \vec{u}\, dS \\ & + \frac{1}{2} \int_{S_{flow}} f(S) \left(\vec{u} - \vec{u}^0\right)^2 dS \end{aligned} \tag{II.A.5}$$

where $f(S)$ is an arbitrary continuous function on the flow-through boundary S_{flow}, $S_{terrain}$ is the terrain surface, and $\mathbf{D}$ is a 3x3 diagonal matrix with non-zero components $\sigma_i^2\, \delta_{i,j}$ ($\delta_{i,j}$ is the Kronecker delta and $\sigma_1 = \sigma_2 = \sigma_H$ and $\sigma_3 = \sigma_V$).

The Euler-Lagrange and Poisson equations are unchanged from the simple CG-MATHEW formulation (Eqs. II.A.2 and II.A.4)

$$\vec{u} - \vec{u}^0 = \mathbf{D} \cdot \nabla\lambda \tag{II.A.6}$$

$$\nabla \cdot \mathbf{D} \cdot \nabla\lambda = -\nabla \cdot \vec{u}^0 \ , \tag{II.A.7}$$

but the boundary conditions now contain contributions from the new surface integrals as well as from the integration by parts

$$\begin{aligned} 0 &= \int_S \lambda \vec{n} \cdot \delta\vec{u}\, dS - \int_{S_{terrain}} \lambda_{terrain}\, \vec{n} \cdot \delta\vec{u}\, dS + \int_{S_{flow}} f(S)\left(\vec{u} - \vec{u}^0\right) \cdot \delta\vec{u}\, dS \\ &= \int_{S_{terrain}} \left(\lambda_{terrain} - \lambda\right)\vec{n} \cdot \delta\vec{u}\, dS + \int_{S_{flow}} \left[f(S)\left(\vec{u} - \vec{u}^0\right) + \lambda\vec{n} \right] \cdot \delta\vec{u}\, dS \ . \end{aligned} \tag{II.A.8}$$

For the first integral to vanish, λ must be equal to $\lambda_{terrain}$ on $S_{terrain}$. The flow-through boundary condition and terrain surface constraint can then be written as

---

[1] The original MATHEW code had OPEN boundaries corresponding to the ADJUST_NORMAL condition and CLOSED boundaries equivalent to FIXED_NORMAL on terrain surfaces with $\vec{n} \cdot \vec{u}^0\big|_{S_{terrain}} = 0$.

$$\begin{aligned} \lambda + f(S)\vec{n} \cdot \left(\vec{u} - \vec{u}^0\right) &= \lambda + f(S)\vec{n} \cdot \mathbf{D} \cdot \nabla\lambda = 0 \qquad S_{flow} \\ \vec{n} \cdot \vec{u} &= \vec{n} \cdot \vec{u}^0 + \vec{n} \cdot \mathbf{D} \cdot \nabla\lambda = 0 \qquad S_{terrain} \end{aligned} \tag{II.A.9}$$

where the Euler-Lagrange equations have been used to substitute for $\vec{u}$. For a rectangular grid, these equations reduce to

$$\begin{aligned} \lambda + \sigma_n^2 \, f(S) \frac{\partial\lambda}{\partial n} &= 0 \qquad S_{flow} \\ \vec{n} \cdot \vec{u}^0 &= -\sigma_n^2 \frac{\partial\lambda}{\partial n} \qquad S_{terrain} \end{aligned} \tag{II.A.10}$$

where $\vec{n}$ is the *outward* normal to the face. Although in principle, $f(S)$ could be any continuous function, in practice, it is taken to be a constant $1/c_n$ governing the ratio of normal to tangential adjustment on each face, in order to limit the danger of overly constraining the problem. The ADJUST_NORMAL and FIXED_NORMAL boundary conditions are limiting cases of the MIXED boundary condition as $c_n \to \infty$ or $c_n \to 0$, respectively.

II.A.3. Derivation of the CG-MATHEW stencil

The CG-MATHEW grid is rectangular with regular spacings of Δx, Δy, and Δz, indexed by *i*, j, and *k*, which run from 1 to $i_{\max}$, $j_{\max}$, and $k_{\max}$, respectively. The wind field components are defined on staggered face centers, interpolated from input grid-point winds, and λ is defined at the cell center as shown in Figure II.A.1. Standard two- and three-point finite difference formulae then give the first derivatives of λ on the staggered face centers and both the second λ derivative and the wind component first derivatives at the cell centers. The topography is specified by cell-centered heights indexed by $\left(i + \frac{1}{2}, j + \frac{1}{2}\right)$.

For computational purposes, the Poisson equation II.A.7 is divided through by σ_H^2 and discretized, resulting in a set of linear equations in the components $\lambda_{i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}}$, which can be written in matrix form as $\mathbf{C}\vec{\lambda} = \vec{\mathbf{b}}$. $\vec{\lambda}$ and $\vec{\mathbf{b}} = -\left(\nabla \cdot \vec{u}_0\right)/\sigma_H^2$ are vectors of length $N = \left(i_{\max} - 1\right) \cdot \left(j_{\max} - 1\right) \cdot \left(k_{\max} - 1\right)$, and $\mathbf{C}$ is an $N \times N$ matrix containing the finite-difference coefficients for each cell, which depend upon the boundary condition. Since the upper boundary of a given cell is the lower boundary of the next higher index cell on that axis, $\mathbf{C}$ is symmetrical, and can be constructed to be positive definite ($\vec{a} \cdot \mathbf{C} \cdot \vec{a} \geq 0$ for all vectors $\vec{a}$) [2].

---

[2] The original MATHEW stencil was derived using quadratic differencing formulae leading to an asymmetric coefficient matrix.

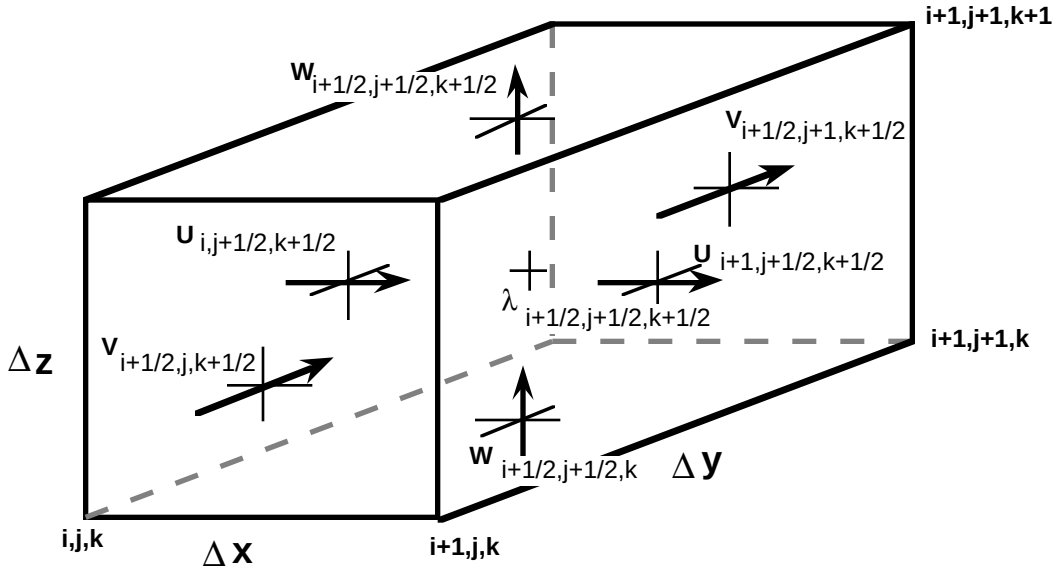


Figure II.A.1: Schematic representation of CG-MATHEW cell centered at point $(i + 1/2, j + 1/2, k + 1/2)$ showing staggered face-centered wind and cell-centered λ components.

The stencil for the coefficient matrix can be derived for any one of the three axis directions, with symmetry used to establish the other components. In the x direction, the boundary conditions on the i and $i + 1$ faces determine relationships between $\lambda_{i-1/2}$, $\lambda_{i+1/2}$, and $\lambda_{i+3/2}$, which are used in the discretized Poisson and Euler-Lagrange equations

$$\frac{\lambda_{i+3/2} - 2\lambda_{i+1/2} + \lambda_{i-1/2}}{(\Delta x)^2} = -\frac{1}{\sigma_H^2}\frac{u_{i+1}^0 - u_i^0}{\Delta x} \quad , \quad u_i = u_i^0 + \frac{\lambda_{i+1/2} - \lambda_{i-1/2}}{\Delta x} \quad . \qquad \text{(II.A.11)}$$

Terrain cells are decoupled by setting the diagonal $\lambda_{i+1/2}$ element to one and the off-diagonal elements of the matrix row to zero, so that the value of λ for a terrain cell is forced to be zero. Bordering interior cells impose no constraints on the λ values, while other boundary conditions restrict the allowed values as summarized in Table II.A.1. The opposite sign on the derivative for the i MIXED boundary condition reflects the reverse direction of the *outward* normal. It should be noted that for ADJUST_NORMAL boundary conditions, the tangential component is algebraically constrained but not fixed, due to the staggered face-centered representation of the wind components. Combinations of the boundary conditions can be treated by a straightforward extension.

CELL $i+\frac{1}{2}$	BOUNDARY i	BOUNDARY $i+1$		
terrain				$\lambda_{i+1/2} = 1$
interior	ADJUST_NORMAL			$\lambda_{i-1/2} = -\lambda_{i+1/2}$
		ADJUST_NORMAL		$\lambda_{i+3/2} = -\lambda_{i+1/2}$
	FIXED_NORMAL			$\lambda_{i-1/2} = \lambda_{i+1/2}$
		FIXED_NORMAL		$\lambda_{i+3/2} = \lambda_{i+1/2}$
	MIXED			$\lambda_{i-1/2} = \left(\frac{1-c_x^- \frac{\Delta x}{2}}{1+c_x^- \frac{\Delta x}{2}}\right)\lambda_{i+1/2}$
		MIXED		$\lambda_{i-3/2} = \left(\frac{1-c_x^+ \frac{\Delta x}{2}}{1+c_x^+ \frac{\Delta x}{2}}\right)\lambda_{i+1/2}$

Table II.A.1: Boundary condition restrictions on λ values. The constant $c_x^{\pm}$ controls the ratio of normal to horizontal adjustment for the MIXED condition.

Due to the symmetric sparse nature of the Poisson matrix, considerable memory savings are possible by storing only the diagonal and sub-diagonal elements. For the three-dimensional case, this can be generalized to storing the four possibly non-zero lower triangle components of **C** in a compact 4 x N form written schematically as

$$\mathbf{A} = \left| \begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ C_{i,j,k-1} & C_{i,j-1,k} & C_{i-1,j,k} & C_{i,j,k} \\ \vdots & \vdots & \vdots & \vdots \end{array} \right| \quad , \tag{II.A.12}$$

where N is the number of cells. The n^{th} or $(i, j, k)^{th}$ element of the matrix multiply $\mathbf{C} \cdot \lambda$ then becomes

$$\begin{aligned} \left(\mathbf{C} \cdot \vec{\lambda}\right)_n &= \left(\mathbf{C} \cdot \vec{\lambda}\right)_{i,j,k} \\ &= A_{i,j,k,4}\,\lambda_{i,j,k} + A_{i,j,k,3}\,\lambda_{i-1,j,k} + A_{i,j,k,2}\,\lambda_{i,j-1,k} + A_{i,j,k,1}\,\lambda_{i,j,k-1} \\ &+ A_{i+1,j,k,3}\,\lambda_{i+1,j,k} + A_{i,j,+1,k,2}\,\lambda_{i,j+1,k} + A_{i,j,k+1,1}\,\lambda_{i,j,k+1} \\ &= A_{n,4}\,\lambda_n + A_{n,3}\,\lambda_{n-1} + A_{n,2}\,\lambda_{n-i_{\max}} + A_{n,1}\,\lambda_{n-i_{\max}\cdot j_{\max}} \\ &+ A_{n+1,3}\,\lambda_{n+1} + A_{n+i_{\max},2}\,\lambda_{n+i_{\max}} + A_{n+i_{\max}\cdot j_{\max},1}\,\lambda_{n+i_{\max}\cdot j_{\max}} \end{aligned} \tag{II.A.13}$$

since $\pm i_{\max}$ moves ± 1 in the j coefficient and $\pm(i_{\max} \cdot j_{\max})$ moves ± 1 in the k coefficient. We will also call this the matrix multiply $\mathbf{A} \cdot \lambda$.

The complete CG-MATHEW stencil in three-dimensions is given in terms of the components of **A** in Table II.A.2. The matrix elements are found by summing together the contributions in each column which match the appropriate boundary conditions. For example, if the cell $n = (i, j, k)$ is bordered by terrain cells at $i-1$ and $j+1$ and by interior cells on the remaining sides, $A_{n,1} = \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$, $A_{n,2} = \frac{1}{(\Delta y)^2}$, $A_{n,3} = 0$, and $A_{n,4} = -\frac{1}{(\Delta x)^2} - \frac{1}{(\Delta y)^2} - \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{2}{(\Delta z)^2}$.

II.A.4. Conjugate Gradient Algorithm

Conjugate gradient methods for solving an N x N linear system $\mathbf{A}\vec{x} = \vec{b}$ have several advantageous features. They require no parameter estimations and provide optimal error minimization. The algorithms have a finite termination after $M \leq N$ steps, where M is the number of distinct eigenvalues of **A**. Since they reference **A** only through vector multiplication, the codes are also highly vectorizable and allow concise storage for sparse systems.

Conjugate gradient algorithms (Varga, 1962) are based on the idea of minimizing a function $f(\vec{x}) = \frac{1}{2}\vec{x}^T \cdot \mathbf{A} \cdot \vec{x} - \vec{x}^T \cdot \vec{b}$ by finding the point where $\nabla f = \mathbf{A} \cdot \vec{x} - \vec{b} = 0$. The minimization is carried out by generating a succession of search directions $\vec{p}_k$ and a corresponding set of improved solutions $\vec{x}_{k+1}$ which minimize f in the space spanned by $\{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_k\}$. Convergence is monotonic, the generalized error function $(A\vec{x} - \vec{z})^T \cdot (\vec{x} - \vec{z})$ taking on its smallest value in the subspace spanned by the search when $\vec{z} = \vec{x}_{k+1}$ and decreasing in magnitude with each iteration.

For improved convergence, preconditioning is used to modify the matrix **A** to one that is very close to the identity matrix **I** except for a few extreme eigenvalues. This is done by multiplying **A** by a easily calculable sparse symmetric matrix $\tilde{\mathbf{A}}^{-1}$

$$\begin{aligned} &\tilde{\mathbf{A}}^{-1}\,\mathbf{A} \approx \mathbf{I} \\ &\tilde{\mathbf{A}}^{-1}\,\mathbf{A}\vec{x} = \tilde{\mathbf{A}}^{-1}\vec{b} \quad . \end{aligned} \tag{II.A.14}$$

As a first approximation, diagonal scaling

$$\tilde{\mathbf{A}}^{-1}_{i,j} = \frac{1}{\mathbf{A}_{i,j}} \tag{II.A.15}$$

is used for preconditioning.

The numerical algorithm (Press et al., 1993) uses an initial guess for $\vec{x}_1$, which is an arbitrary approximation to the solution vector and may be taken to be the zero vector. The first residual $\vec{r}_1$, the preconditioned residual $\vec{z}_1$, and an initial direction

CELL	BOUNDARY		$A_{n,1}$	$A_{n,2}$	$A_{n,3}$	$A_{n,4}$
terrain	all boundaries					1
interior	all boundaries					$-\frac{2}{(\Delta x)^2}-\frac{2}{(\Delta y)^2}-\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{2}{(\Delta z)^2}$
	x_{i-1}	interior			$\frac{1}{(\Delta x)^2}$	
		$\lambda = 0$				$-\frac{1}{(\Delta x)^2}$
		$\frac{\partial\lambda}{\partial x}=0$				$\frac{1}{(\Delta x)^2}$
		$c_x^-\lambda-\frac{\partial\lambda}{\partial x}=0$				$\frac{1-c_x^-\Delta x/2}{1+c_x^-\Delta x/2}\cdot\frac{1}{(\Delta x)^2}$
	x_{i+1}	interior				
		$\lambda = 0$				$-\frac{1}{(\Delta x)^2}$
		$\frac{\partial\lambda}{\partial x}=0$				$\frac{1}{(\Delta x)^2}$
		$c_x^+\lambda+\frac{\partial\lambda}{\partial x}=0$				$\frac{1-c_x^+\Delta x/2}{1+c_x^+\Delta x/2}\cdot\frac{1}{(\Delta x)^2}$
	y_{j-1}	interior		$\frac{1}{(\Delta y)^2}$		
		$\lambda = 0$				$-\frac{1}{(\Delta y)^2}$
		$\frac{\partial\lambda}{\partial y}=0$				$\frac{1}{(\Delta y)^2}$
		$c_y^-\lambda-\frac{\partial\lambda}{\partial y}=0$				$\frac{1-c_y^-\Delta y/2}{1+c_y^-\Delta y/2}\cdot\frac{1}{(\Delta y)^2}$
	y_{j+1}	interior				
		$\lambda=0$				$-\frac{1}{(\Delta y)^2}$
		$\frac{\partial\lambda}{\partial y}=0$				$\frac{1}{(\Delta y)^2}$
		$c_y^+\lambda+\frac{\partial\lambda}{\partial y}=0$				$\frac{1-c_y^+\Delta y/2}{1+c_y^+\Delta y/2}\cdot\frac{1}{(\Delta y)^2}$
	z_{k-1}	interior	$\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$			
		$\lambda=0$				$-\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$
		$\frac{\partial\lambda}{\partial z}=0$				$\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$
		$c_z^-\lambda-\frac{\partial\lambda}{\partial z}=0$				$\frac{1-c_z^-\Delta z/2}{1+c_z^-\Delta z/2}\cdot\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$
	z_{k+1}	interior				
		$\lambda=0$				$-\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$
		$\frac{\partial\lambda}{\partial z}=0$				$\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$
		$c_z^+\lambda+\frac{\partial\lambda}{\partial z}=0$				$\frac{1-c_z^+\Delta z/2}{1+c_z^+\Delta z/2}\cdot\left(\frac{\sigma_V}{\sigma_H}\right)^2\frac{1}{(\Delta z)^2}$

Table II.A.2: CG-MATHEW stencil. The values of the stencil for any cell n are found by summing contributions from all applicable rows.

vector $\vec{p}_1$ are generated according to

$$\begin{aligned} \vec{r}_1 &= \left(\vec{b} - \mathbf{A}\vec{x}_1\right) \\ \vec{z}_1 &= \tilde{A}^{-1}\,\vec{r}_1 \\ \vec{p}_1 &= \vec{z}_1 \end{aligned} \qquad \text{(II.A.16)}$$

The minimization then proceeds for successive k by finding the α_k that minimize $f\left(\vec{x}_k + \alpha_k \vec{p}_k\right)$ and generating the next search direction vector by **A**-orthogonalization of the residuals according to the prescription

$$\begin{aligned} \alpha_k &= \frac{\vec{r}_k \cdot \vec{z}_k}{\vec{p}_k \cdot \mathbf{A} \cdot \vec{p}_k} \\ \vec{r}_{k+1} &= \vec{r}_k - \alpha_k \cdot \mathbf{A} \cdot \vec{p}_k \\ \vec{z}_{k+1} &= \tilde{\mathbf{A}}^{-1} \cdot \vec{r}_{k+1} \\ \vec{x}_{k+1} &= \vec{x}_k - \alpha_k \vec{p}_k \\ \beta_k &= \frac{\vec{r}_k \cdot \vec{z}_{k+1}}{\vec{r}_k \cdot \vec{z}_k} \\ \vec{p}_{k+1} &= \vec{z}_k - \beta_k \vec{p}_k \end{aligned} \qquad \text{(II.A.17)}$$

The residuals and direction vectors are mutually orthogonal and conjugate

$$\begin{aligned} \vec{r}_i \cdot \vec{r}_j &= 0 \quad j < i \\ \vec{r}_i \cdot \vec{p}_j &= 0 \quad j \le i \\ \vec{p}_i \cdot \mathbf{A} \cdot \vec{p}_j &= 0 \quad j < i \end{aligned} \qquad \text{(II.A.18)}$$

so that each iteration proceeds by making the residual orthogonal to one more independent conjugate direction vector. The residuals $\vec{r}_k$ are also linearly independent and give the direction of steepest descent $f(\vec{x})$ at each step. It should be noted that the conjugate direction vectors $\vec{p}_k$ are *not* determined beforehand. Instead, they are constructed as linear combinations of the $\vec{r}_k$ by calculation of the appropriate β_k at each iteration.

Any of several convergence criterion can be adopted depending on the application. We have chosen a combination of

$$\begin{aligned} \left|\mathbf{A}\vec{x}_k - \vec{b}\right| &< \Delta_{tolerance} \\ \frac{\left|\vec{x}_{k+1} - \vec{x}_k\right|}{\left|\vec{x}_{k+1}\right|} &< \Delta_{tolerance} \end{aligned} \qquad \text{(II.A.19)}$$

The first tolerance check is used to stop the iterations, whenever the second condition is satisfied. These criteria are not completely robust; the final field should only be trusted when the solution converges monotonically.

II.A.5. Output

The face-centered adjusted velocity components (u, v, w) are interpolated back to form grid-point winds using an averaging procedure which maintains mass-consistency. The grid-point velocities are then written to MATVEL files (see Section III.C.8) along with traceback black information which uniquely identifies the current problem. These files provide input for the pollutant transport and diffusion code ADPIC.

II.B. ADPIC

II.B.1. Advection-Diffusion Equation and its Lagrangian Equivalent

The Random Displacement Method (RDM), as well as the original particle-in-cell, gradient diffusion method used in ADPIC, is based upon the conservation of species principal expressed in the form of the 3-D, incompressible, Eulerian advection-diffusion equation:

$$\frac{\partial \overline{C}}{\partial t} = -\overline{u}\frac{\partial \overline{C}}{\partial x} - \overline{v}\frac{\partial \overline{C}}{\partial y} - \overline{w}\frac{\partial \overline{C}}{\partial z} + \frac{\partial}{\partial x}\left(K_x \frac{\partial \overline{C}}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial \overline{C}}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial \overline{C}}{\partial z}\right) \qquad \text{(II.B.1)}$$

where $\overline{C}$ is the mean air concentration of the species; $\overline{u}$, $\overline{v}$, and $\overline{w}$ are the mean wind components in the x, y, and z directions, respectively; t is time; and K_x, K_y, and K_z are the eddy diffusivities for the three coordinate directions (the eddy diffusivity tensor is assumed to be diagonal).

The RDM uses stochastic differential equations that describe the same process as the advection-diffusion equation (II.B.1), but in a Lagrangian instead of an Eulerian framework (Boughton *et al.*, 1987; Ermak, 1992; Rodean *et al*., 1992). The stochastic differential equations for the displacement of an idealized fluid element in the three coordinate directions are

$$dx = \overline{u}dt + (2K_x)^{1/2} dW_x, \qquad \text{(II.B.2a)}$$

$$dy = \overline{v}dt + (2K_y)^{1/2} dW_y, \text{ and} \qquad \text{(II.B.2b)}$$

$$dz = \overline{w}dt + \frac{\partial K_z}{\partial z}dt + (2K_z)^{1/2} dW_z, \qquad \text{(II.B.2c)}$$

where $dW_{x,y,z}$ are three independent random variates with zero mean and variance dt, i.e.,

$$\overline{dW} = 0,$$

$$\overline{dW^2} = dt.$$

In (II.B.2a-b) it has been assumed that turbulence is homogeneous in the horizontal, x and y, directions. Eqs. (II.B.2a-c) are integrated in time to calculate independent trajectories representing the movement of individual fluid elements. In a RDM calculation, the released species is represented by a large number of marker particles (fluid elements), each with a specified species mass. The ensemble-mean concentration at any time t can then be calculate from the particle locations at time t and the species mass associated with each particle.

The particle-in-cell, gradient diffusion method of solving the advection diffusion equation is a hybrid Eulerian-Lagrangian pseudo-velocity method developed by Lange (1973, 1978, and 1989), and described in earlier versions of this user's guide. The gradient diffusion method will *not* be described further in this chapter. The DIFF_METH parameter in the ADPIC_CONTROL namelist controls which diffusion method is used: RDM or gradient diffusion.

A fundamental assumption in both the Eulerian formulation (II.B.1) and the Lagrangian formulation (II.B.2a-c) is that time and space scales of the turbulence are small relative to the time and space scales of the diffusion calculation to be performed. From an Eulerian view, the length scale ("eddy size") of the turbulence, L, is assumed to be small relative to the length scale of the spatial gradients of the concentration to be calculated, e.g., $L << \overline{C} \Big/ \left(\frac{\partial \overline{C}}{\partial z} \right)$. From a Lagrangian view, the time scale ("eddy lifetime") of the turbulence, τ, is assumed to be smaller than the time over which concentrations are to be calculated (i.e., $\tau << t$).

In the atmospheric boundary layer, turbulent time scales range from seconds to minutes. At the surface, they are typically assumed to be zero and to increase in a manner that is approximately linear with height. The rate of increase with height of the turbulent time scale near the surface is much higher under unstable, as opposed to stable, conditions. Therefore, for near-source diffusion from a point source the assumptions implicit in eqs. (1) and (2a-c) become more valid, in general, for source heights closer to the surface and for more stable conditions. Conversely, for more elevated source heights and more unstable conditions (when the time and space scales of turbulence are greater) the assumptions are only valid for times and distances farther from the source, when the concentration distribution is more dispersed.

II.B.2. RDM Numerical Solution

In the RDM, it is assumed that K_x and K_y are constant in space, and that K_z varies linearly with height, i.e.,

$$K_z(z) = K_z(z_i) + \left. \frac{\partial K}{\partial z} \right|_{z_i} \left(z - z_i \right). \tag{II.B.3}$$

Then, a numerical solution of eqs. (II.B.2a-c) can be constructed using the difference equations (Ermak, 1992)

$$\Delta x_i = \overline{u} \Delta t_i + \left[2K_{xi} \Delta t_i \right]^{1/2} \xi_{xi}, \tag{II.B.4a}$$

$$\Delta y_i = \overline{v} \Delta t_i + \left[2K_{yi} \Delta t_i \right]^{1/2} \xi_{yi}, \text{ and} \tag{II.B.4b}$$

$$\Delta z_i = \overline{w} \Delta t_i + \left(\frac{\partial K_z}{\partial z} \right)_i \Delta t_i + \left[2K_{zi} \Delta t_i + \left(\frac{\partial K_z}{\partial z} \right)_i^2 \Delta t_i^2 \right]^{1/2} \xi_{zi}, \tag{II.B.4c}$$

where, for example, in the vertical direction,

$$\Delta z_i = z(t_{i+1}) - z(t_i),$$
$$K_{zi} = K_z(z_i, t_i),$$
$$\left(\frac{\partial K_z}{\partial z}\right)_i = \left.\frac{\partial K}{\partial z}\right|_{\substack{t=t_i \\ z=z_i}},$$
$$\Delta t_i = t_{i+1} - t_i, \text{and}$$
$$\xi_{zi} = \text{ a random number with zero mean } (\overline{\xi} = 0)$$
$$\text{and variance of one } (\overline{\xi^2} = 1).$$

The second-order term in (II.B.4c), $(\partial K/\partial z)^2 \Delta t^2$, is unique to our RDM. This term includes the effect of the linear variation of K during the finite time step, Δt, on the displacement variance, and is especially important close to the ground where K_z approaches zero in similarity theory parameterizations. Eddy diffusivity and the spatial derivative of the vertical eddy diffusivity are evaluated analytically at the particle position at each time step. Particle trajectories can be numerically calculated by successively using (II.B.4a-c) and incrementing the particle position using, for example,

$$z(t_{i+1}) = z(t_i) + \Delta z_i. \qquad \text{(II.B.5)}$$

Details on the implementation of (II.B.4-5) are given by Ermak *et al.* (1995).

In the ADPIC code, a gridded wind field is used for the mean wind components, $\overline{u}$, $\overline{v}$, and $\overline{w}$ in (II.B.4a-c). The time step used in the numerical solution is restricted to prevent a particle from moving more than a single mean wind grid cell in any time step due to mean wind advection plus diffusion. Furthermore, for accurate time integration of concentration in the concentration grid, the time step is also restricted to prevent any particle from moving through more than a single concentration grid cell in any time step (this results in conservatively small time steps when the concentration distribution spans many concentration sampling grid cells).

The *local time step* is determined separately for each particle. The local time steps for all particles are forced to synchronize at necessary times, such as when meteorological parameters change, source parameters change, wind input data changes, particle concentrations are to be saved, status messages are to be written, or graphics output is to be written. The *global time step* is the time interval between these events that affect all particles. The local time step cannot exceed the global time step. Time steps reported to the user through the status messages to the monitor or the log file are the global timesteps along with the average of the local particle time steps during each global time step.

II.B.3. Turbulence Parameterizations

As can be seen in the previous section, the RDM used in ADPIC requires parameterizations of atmospheric turbulence in the form of an eddy diffusivities, K. Several K parameterization options are available for use with the RDM. Turbulence parameterizations for the Gradient Diffusion

method are not described here, but are described in chapter III and in previous versions of this user's guide.

Tables II.B.3.a and II.B.3.b summarize the options available for both vertical and horizontal turbulence, respectively, in the RDM option in ADPIC. In the RDM option, two parameters, TURB_PARAM_VERT and TURB_PARAM_HORZ, control the vertical and horizontal turbulence parameterizations separately. These options are discussed in the following section II.B.3 and in section III.B.9 which describes the ADPIC namelist input parameters. Further information on these turbulence parameterizations is given by Nasstrom (1995).

In the Gradient Diffusion option, which is not described here, the parameter TURB_PARAM_TYPE, controls the turbulence parameterization in the vertical and horizontal simultaneously. See section III.B.9 for a description of TURB_PARAM_TYPE.

Table II.B.3.a. Vertical turbulence options in the RDM

TURB_PARAM_VERT=	**Description**
'KZ_SIMTHRY'	K_z as a function of z; calculated from boundary layer similarity theory as a function of u_*, h, z and L for $z < h$. Constant, K_z^{tropo}, used for $z > h$.
'KZ_SIMTHRY_USER'	K_z as a function of z; calculated from boundary layer similarity theory as function of u_*, h, z and L, but with *user-specified* coefficients a, b, c, c_2, c_3, and K_z^{tropo}
'SIGZ_PWRLAWX_PG'	K_z as a function of downwind distance, x; calculated from *Pasquill-Gifford* $\sigma_z(x)$ in a power law form ($\sigma_z = \gamma x^{\rho}$).
'SIGZ_PWRLAWX_USER'	K_z as a function of x; calculated from the power-law form of $\sigma_z = \gamma x^{\rho}$ and *user-specified* coefficients γ and ρ.
'KZ_LINEARZ'	K_z calculated from a linear function of height: $K_z = K_{z\mathrm{ref}}\left(z/z_{\mathrm{ref}}\right)$, where K_{zref} and z_{ref} are specified by the user.
'KZ_CONSTANTZ'	K_z fixed at constant value.

Table II.B.3.a. Horizontal turbulence options in the RDM

TURB_PARAM_HORZ=	Description
'SIGH_SIGTHETA'	K_H as a function of particle age (time) *t;* calculated from *Draxler's* $\sigma_y(t)$ using input σ_θ data and internal-default τ_H
'SIGH_SIGTHETA_USER'	K_H as a function of particle age (time) *t*; calculated from *Draxler's* $\sigma_y(t)$ using input σ_θ data and *user-specified* τ_H.
'SIGH_PWRLAWX_PG'	K_H as a function of downwind distance *x*; calculated from *Pasquill-Gifford* $\sigma_y(x)$ in a power law form ($\sigma_y = \alpha x^\beta$).
'SIGH_PWRLAWX_MESO'	K_H as a function of *x;* calculated from *MESODIF* $\sigma_y(x)$.
'SIGH_PWRLAWX_LONGRANGE'	K_H as a function of *x*; calculated from *Rodriguez's long-range* $\sigma_y(x)$.
'SIGH_PWRLAWX_USER'	K_H as function of *x*; calculated from power law form of $\sigma_y = \alpha x^\beta$ and *user-specified* coefficients α and β.

Preferred parameterizations are based on boundary layer turbulence scaling parameters and measured turbulent velocity statistics. Simpler parameterizations, based solely on Pasquill stability class, are also available. When eddy diffusivities are based on boundary layer turbulence scaling parameters (i.e., friction velocity,u_*; mixed layer height, *h*; height, *z*; and Obukhov length, *L*), "turbulence parameterization" is an appropriate term. In other cases, this term is used loosely to describe "sigma curves". These are semi-empirical relationships between the standard deviations, $\sigma_z(x)$ and $\sigma_y(x)$, of concentration from a point source and downwind distance. Separate sigma curves are used for each of six Pasquill stability classes, which are used to categorize the diffusive properties of the atmospheric surface layer. Consequently, sigma curves are more than parameterizations of turbulence since they also prescribe the final concentration distribution (for a point source) given a Pasquill stability class.

In the ADPIC model, sigma curves can be used to calculate the vertical and horizontal eddy diffusivities, K_z and K_H ($= K_x = K_y$). Thus, they can be used to "back out" parameterizations for *K* which are consistent with the dispersion associated with the particular sigma curve. This results in

eddy diffusivities which are spatially homogeneous, but travel time dependent. A vertically homogeneous K_z is a poor assumption in the atmospheric boundary layer. A horizontally homogeneous K_H is a better assumption because quasi-horizontally-homogeneous turbulence is more common in the atmospheric boundary layer. The use of these time-dependent eddy diffusivities is an empirical approach to calculating diffusion at short times when the assumptions implicit in the advection-diffusion equation (on which the RDM is based) are less valid.

An improvement to the overly-broad, site-dependent stability class/sigma curve approach is the use of direct statistical measures of turbulent fluctuations (i.e., σ_θ) along with a horizontal turbulence time scale, τ_H, to determine a time-dependent horizontal concentration standard deviation, $\sigma_y(t)$. These are used in the RDM as a semi-empirical method of parameterizing a time-dependent, horizontally-homogeneous K_H.

II.B.3.a. Vertical Turbulence Parameterizations

TURB_PARAM_VERT = 'KZ_SIMTHRY' AND 'KZ_SIMTHRY_USER'

These options use K_z profiles based on surface layer and boundary layer scaling parameters (Lange, 1989). For the boundary layer ($z < h$), K_z^{bl} is a function of z, L, h, and u_* as follows:

$$K_z^{bl}(z) = \frac{k u_* z}{\phi_z} e^{-cz/h}, \tag{II.B.6}$$

where

$$\phi_z = \left(1 + a\frac{z}{L}\right)^b, \tag{II.B.7}$$

$$a = \begin{cases} -15, & \text{if } 1/L < 0 \\ 0, & \text{if } 1/L = 0 \\ 4.5, & \text{if } 1/L > 0 \end{cases}$$

$$b = \begin{cases} -0.25, & \text{if } 1/L < 0 \\ 1, & \text{if } 1/L \geq 0 \end{cases}, \text{ and}$$

$$c = 4.$$

These default values for a, b, and c can be overridden by user input.

For the free troposphere ($z > h$), K_z is constant with height with a value of

$$K_z = K_z^{tropo}. \tag{II.B.8}$$

In order to obtain a single K_z function for all z which is a smooth, differentiable function of z and which provides a quick transition from K_z^{bl} to K_z^{tropo} near $z=h$, a function $H(z)$ proposed by Ermak (1994, private communication) is used as follows:

$$K_z = H(z)K_z^{bl}(z) + [1 - H(z)]K_z^{tropo}, \tag{II.B.9}$$

where

$$H(z) = e^{-\left(z/c_2 h\right)^{c_3}}, \tag{II.B.10}$$

K_z^{tropo} may be larger or smaller than the value of K_z^{bl} at the mixed layer height, h. The current default values for the constants in Eqs. (II.B.9-10) are K_z^{tropo} = 0.01 m^2/s, $c_2 = 1.5$, and $c_3 = 8$. The user may specify values for all the constants in the K_z parameterization using the 'KZ_SIMTHRY_USER' option. Fig. II.B.1 shows an example K_z profile for a neutral stability case.

Since this turbulence parameterization uses fundamental boundary layer turbulence scaling parameters and accounts for vertical inhomogeneity of K_z, it is the preferred option for vertical turbulence parameterization.

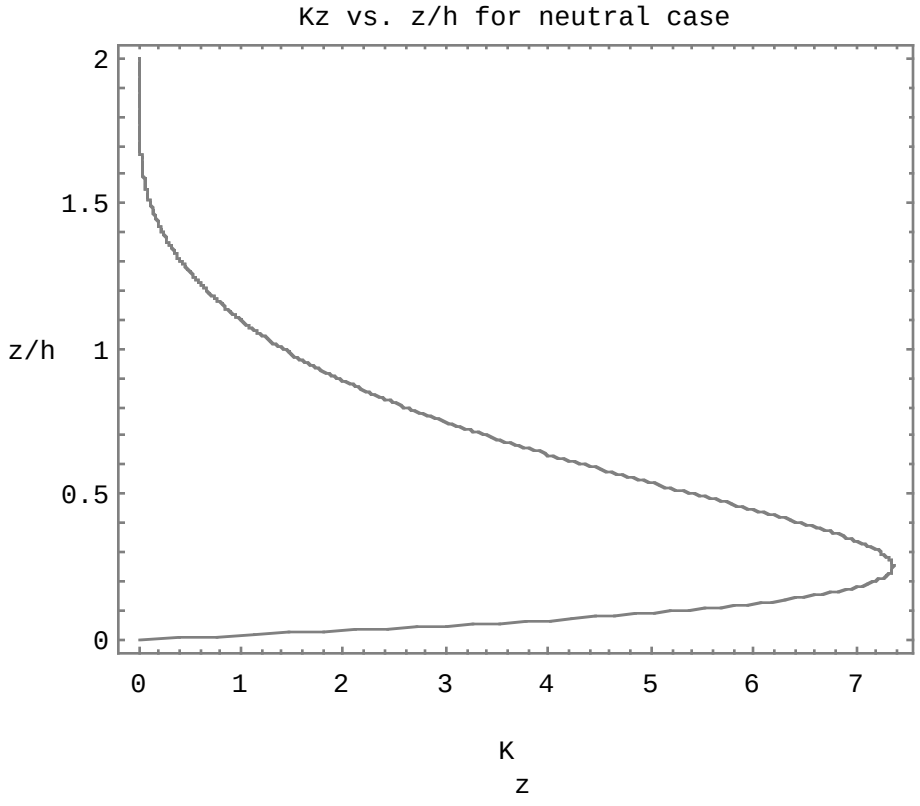


Fig. II.B.1 Example K_z profile for a neutral stability case (u_* = 0.4 m/s, h = 500 m, $1/L$ = 0, K_Z^{tropo} = 0.01 m^2/s).

TURB_PARAM_VERT = 'SIGZ_PWRLAWX_PG'

In this option K_z is determined from Pasquill-Gifford $\sigma_z(x)$ using only stability class. K_z is calculated from $\sigma_z(x)$ in the following form:

$$\sigma_z = \gamma x^{\rho}. \tag{II.B.11}$$

There are different coefficients γ and ρ for each of six Pasquill stability classes, A, B, C, D, E, and F, corresponding in the ADPIC code to stab_class = 1, 2, 3, 4, 5 and 6, respectively. Values for these coefficients were determined by Lange (1993, private communication) by fitting sigma curves developed by Pasquill, adapted by Gifford, and published in Fig. 3.11 of *Meteorology and Atomic Energy* (Slade, 1968). These curves are based on data from experiments on surface releases with release durations and averaging times of 3-15 min. Therefore, it is recommended for calculations of near-surface sources with release durations, concentration averaging times and wind data temporal resolution close to this 3-15 min time period, and when only stability class is available.

Using the relationship between the variance of the concentration distribution and the eddy diffusivity*,

$$K = \frac{1}{2}\frac{d\sigma^2}{dt}, \tag{II.B.12}$$

and the relationship between mean wind speed (assumed to be constant) and downwind distance#,

$$x = \bar{u}\, t \tag{II.B.13}$$

a relationship for K_z can be obtained:

$$K_z = \gamma^2 \rho \bar{u} x^{(2\rho-1)}. \tag{II.B.14}$$

Values of the γ and ρ coefficients were determined by Lange (1993, private communication) by fitting the close-in portions of the Pasquill-Gifford curves and are as follows:

---

* This relationship assumes K is spatially homogeneous. However, this assumption is not generally valid in the vertical direction in the boundary layer, especially under unstable conditions. Correspondingly, vertical concentration distributions are not Gaussian in the vertical, as is assumed in this option.

\# In calculating K, the time t is the age of the particle since it was emitted from the source, and $\bar{u}$ is the mean wind at the particle location.

stab_class	γ	ρ
1	0.013	1.5
2	0.039	1.2
3	0.222	0.81
4	0.382	0.64
5	0.418	0.58
6	0.288	0.55

A new parameter, $x_{\max}$, specifies the downwind distance beyond which the value of K_z is held constant at its value at that maximum distance. The default values are as follows:

stab_class	$x_{\max}$ (m)
1	5.E+3
2-6	1.E+4

TURB_PARAM_VERT = 'KZ_LINEARZ'

This new option allows K_z to be calculated as a function of height from the linear equation:

$$K_z = K_{z_{\text{ref}}}\left(z/z_{\text{ref}}\right), \tag{II.B.15}$$

where $K_{z_{\text{ref}}}$ is the value of K_z at reference height z_{ref}. $K_{z_{\text{ref}}}$ and z_{ref} must be input by the user, and do *not* have any internal default values. When used with the wind speed power law option, this linear K_z option allows RDM calculations to be compared to analytical solutions to the advection-diffusion equation.

TURB_PARAM_VERT = 'KZ_CONSTANTZ'

This new option allows K_z to be fixed at a constant value, unchanging with height, time or downwind distance:

$$K_z = K_{z_{\text{constant}}}.$$

When used with the constant wind speed option, this constant *K* option allows RDM calculations to be compared to a Gaussian solution to the advection-diffusion equation.

II.B.3.b. Horizontal Turbulence Parameterizations

TURB_PARAM_HORZ='SIGH_SIGTHETA' AND 'SIGH_SIGTHETA_USER'

K_H for this option is based on Draxler's (1976) semi-empirical relationship for σ_y as a function of time and the standard deviation of cross-wind velocity component (σ_v). Draxler used an equation of the following form for σ_y:

$$\sigma_y = \sigma_v\, t\, f\left(t/\tau_H\right) \tag{II.B.16}$$

where

$$f\left(t/\tau_H\right) = \frac{1}{1 + d\left(t/\tau_H\right)^{1/2}}.$$

This formulation results in $\sigma_y \propto t$ for $t << \tau_H$ and $\sigma_y \propto t^{1/2}$ for $t >> \tau_H$. The constant d is a combination of constants used by Draxler. A value of d =0.36 resulted in Draxler's best fit to close-in diffusion data, and is similar to the value of 0.4 recommended by Hanna et al. (1982). Assuming d =0.36, Draxler recommended values of τ_H =50 sec for surface releases and 160 sec for elevated releases (note: Draxler's T_i is equal to $6.36\tau_H$). The default values in the RDM are

d = 0.36 and

τ_H = 50 sec.

The value of τ_H can be overridden by user input. Using Eq. (II.B.12), assuming K_H=K_y, and using the approximation $\sigma_v \approx \sigma_\theta \bar{u}$, the following equation for K_H can be derived:

$$K_H = \frac{\bar{u}^2 \sigma_\theta^{\,2} t\left[1 + \frac{d}{2}\left(\frac{t}{\tau_H}\right)^{1/2}\right]}{\left[1 + d\left(\frac{t}{\tau_H}\right)^{1/2}\right]^3}. \tag{II.B.17}$$

(Note: σ_θ is in radians).

Since this turbulence parameterization uses measured wind direction fluctuations, it is the preferred option for horizontal turbulence parameterization.

TURB_PARAM_HORZ = 'SIGH_PWRLAWX_PG', 'SIGH_PWRLAWX_MESO', 'SIGH_PWRLAWX_LONGRANGE', and 'SIGH_PWRLAWX_USER'

For these options, K_H is based only on stability class and is calculated using $\sigma_y(x)$ in the following form:

$$\sigma_y = \alpha x^{\beta}. \tag{II.B.18}$$

Using Eqs. (II.B.12) and (II.B.13), and assuming again that $K_H = K_y$, the following relationship for K_H can be obtained:

$$K_H = \alpha^2 \beta \bar{u} x^{(2\beta-1)}. \tag{II.B.19}$$

For values of $x \geq x_{\max}$, the value of K_H is held constant at its value at that maximum distance.

The constants α and β depend on the turb_param_horz option used and are described below:

α and β for TURB_PARAM_HORZ = 'SIGH_PWRLAWX_PG':

The α and β coefficients for $\sigma_y(x)$ used in this option are based on a least-squares fit (Lange, 1993, private communication) to the sigma curves developed by Pasquill, adapted by Gifford, and published in Fig. 3.10 of *Meteorology and Atomic Energy* (Slade, 1968). These curves are based on data from experiments involving surface releases with release durations and averaging times of 3-15 min. Therefore, this turbulence parameterization option is recommended for calculations of near-surface releases with release durations, concentration averaging times and wind data temporal resolution close to this 3-15 min time period, and when only stability class is available.

There are separate curves for each of six Pasquill stability classes (A, B, C, D, E, and F, corresponding to stab_class = 1, 2, 3, 4, 5 and 6, respectively). The $\sigma_y(x)$ equation coefficients for this option are

$\beta = 0.92$ and

α, depending on stability class, as follows:

stab_class	α (Pasquill-Gifford)
1	0.42
2	0.30
3	0.22
4	0.15
5	0.09
6	0.06

Default value of x_{max} for this option is 2.E+4 m.

α and β for TURB_PARAM_HORZ = 'SIGH_PWRLAWX_MESO':

The equations for $\sigma_y(x)$ in this parameterization option are based on a fit to experimental data from near-surface releases from the Idaho National Reactor Testing Station (NRTS) (Yanskey et al., 1966) which were subsequently used in the MESODIF model (Start and Wendell, 1974) for the NRTS. These data show increased horizontal diffusion when compared to the Pasquill-Gifford values above, especially under stable conditions. This is due to both (a) the longer release durations (and, therefore, longer averaging times) used in the NRTS data (15-60 min) compared to the release durations in the Pasquill-Gifford data (3-15 min), and (b) the presence of meandering conditions during stable conditions. Therefore, this option is recommended for calculations of near-surface releases with release durations, concentration averaging times and wind data temporal resolution close to this 15-60 min time period, and when when only stability class is available. The $\sigma_y(x)$ equation coefficient values are

β = 0.85 and

α, depending on stability class, as follows:

stab_class	α (MESODIF)
1	0.72
2	0.42
3	0.35
4	0.27
5	0.30
6	0.40

Default value of x_{max} for this option is 2.E+4 m.

α and β for TURB_PARAM_HORZ = 'SIGH_PWRLAWX_LONGRANGE':

In this option, the equations for $\sigma_y(x)$ are based on a fit by Rodriguez *et al.* (1995) to long-range dispersion data. The $\sigma_y(x)$ equation coefficient values are

$$\beta = 1.0,$$

$$\alpha = 0.056, \text{ and}$$

$$x_{max} = 4.E+7 \text{ m}$$

The large default value of 40,000 km for x_{max} is used to assure that these α and β coefficients are, for practical purposes, always used, and, therefore, $\sigma_y \propto t$ essentially for all time. This parameterization reflects the effect of processes other than turbulent diffusion, i.e., spatial and temporal variation in the mean wind. Therefore, this parameterization is recommended for large-scale calculations using wind fields with low spatial resolution (in particular, low vertical resolution so that vertical wind shear in the lowest kilometer is unresolved) and/or low temporal resolution (e.g., 12 hour resolution).

II.B.4. Options in ADPIC

II.B.4.a. Source Parameters

Up to MAXSOURC separate pollutant source configurations (combinations of source type, rate, geometry, etc.) are allowed. The release rate and source center height of each pollutant source can vary with time. The effective times of the time-varying source parameters (SOURCE_RATE and CENTER_HGT) are controlled by the timing parameters SOURCE_START_DATE and SOURCE_START_TIME.

The default method used to create marker particles in a computational interval allocates marker particles proportional to the fraction of the total activity/mass of the total for this source to be released in the interval. An alternative method is also available which creates marker particles proportional to the fraction of the total time that the source is "on" (regardless of the source rate). The first method is especially useful for substantially varying source rates, such as an injection of a large mass of material within a short period of time followed by a relatively much smaller mass of material released over a relatively long period of time. Although the first method is generally recommended, the user should be aware that while the particle statistics will be improved during the peak release time, they will also be degraded during the non-peak release times. The input parameter PART_GEN_MODE is used to control which method is used.

The way in which ADPIC describes pollutants in terms of discrete mass (activity) particles makes the implementation of decay rate calculations straightforward. Such processes as radioactive decay of pollutants in conjunction with atmospheric dispersion under complex terrain conditions can be modeled.

II.B.4.b. Source Geometry

The initial position (x, y, and z coordinates) of each marker particle representing the pollutant material is computed by using a Gaussian or normal probability distribution function. The probability of the distance that each particle will be from each of the three orthogonal axes through the source center is determined by the shape of a Gaussian distribution. The number of particles to be generated for each source is specified by the user by the use of the TOTPART parameter. Several thousand particles per ADPIC source provide a fairly good representation of a sample population.

In all, ten parameters are used to control the representation of the initial stabilized source geometry. The first three are standard deviations of the Gaussian distribution, SIGX, SIGY and SIGZ, in the x, y, and z directions, respectively. These control how peaked or flat the source distribution will be along each axis, as shown in Fig. II.B.2. The Gaussian distributions are then truncated as shown in Fig. II.B.3 using six source cutoff parameters, XPOS_CUTOFF, XNEG_CUTOFF, YPOS_CUTOFF, YNEG_CUTOFF, ZPOS_CUTOFF, and ZNEG_CUTOFF, where POS is for right or top, NEG for left or bottom. The last source geometry parameter is the height above ground of the source center, CENTER_HGT.

If very little truncation of the Gaussian curves is desired, the radius of the entire source is typically used for the cutoff values and the radius divided by 2.5 to compute the standard deviations. This is based on the fact that about 99% of the area in the Gaussian distribution lies within 2.5 standard deviations of its center.

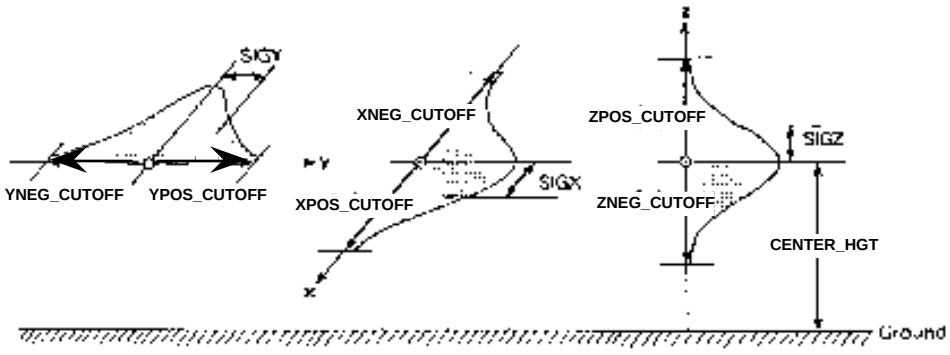


Fig. II.B.2. Gaussian source geometry parameters

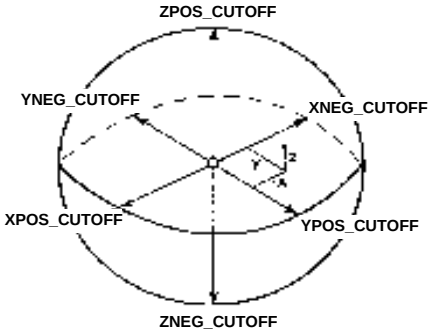


Fig. II.B.3. Cutoff parameters for a spherical-shaped source

The ten source parameters may be used to describe a wide variety of sources, with the three most common shown in Fig. II.B.4. Releases from stacks are usually assumed to be spherical (equal sigma and cutoff values in each direction). Puffs or clouds from fires or explosions are usually elongated in the vertical. If necessary an explosion may be described by two source terms, one for the lower stem and one for the upper cloud.

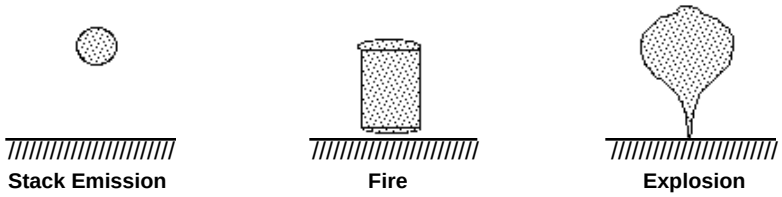
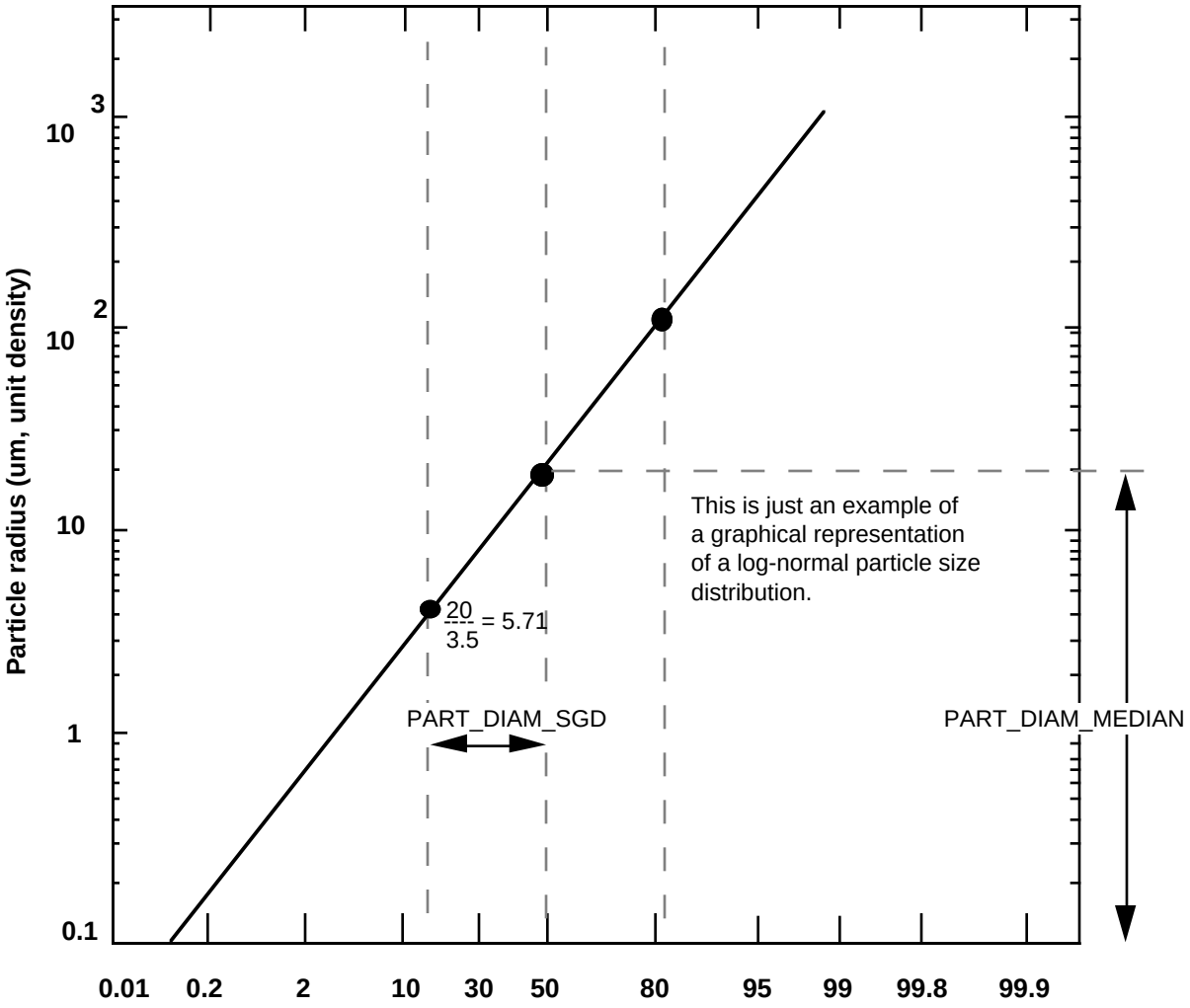


Fig. II.B.4. Example source geometries

II.B.4.c. Particle Size Distribution

The distribution of the particle size range as a log-normal distribution can be specified in ADPIC by using the parameters PART_DIAM_MIN, PART_DIAM_MAX, PART_DIAM_SGD, and PART_DIAM_MEDIAN. An example showing the graphical derivation of these values is shown in Fig. II.B.5. Usually particle sizes are normalized to a particle density of 1 g/cm3 (1000 kg/m3). If actual particle sizes are used, then the actual particle density (PART_DENSITY) must be specified.



Log-normal particle size parameters for ADPIC_SOURCE namelist

Input parameters	Description (Percent less than stated size)
PART_DIAM_SGD	Standard geometric deviation: the ratio of $\frac{\text{radius 50\%}}{\text{radius 16\%}}$ or $\frac{\text{radius 84\%}}{\text{radius 50\%}}$
PART_DIAM_MEDIAN (μm)	Median diameter: diameter of 50% point
PART_DIAM_MAX (μm)	Maximum diameter of particles
PART_DIAM_MIN (μm)	Minimum diameter of particles

Fig. II.B.5. Log normal particle size distribution

II.B.4.d. Dry Deposition

The removal of material from the atmosphere through dry deposition is modeled separately by ADPIC via two processes: gravitational settling of material based on particle size, and flux of material from the surface layer to the ground surface based on a specified deposition velocity.

The settling velocity is modeled by ADPIC with either the Stokes law or the McDonald method (McDonald, 1960) based on particle size (PART_DIAM_MEDIAN), particle density (PART_DENSITY), and altitude. The McDonald method is used for cases when the Reynold's number is greater than 1 and the Stokes Law is invalid (e.g. particle diameters greater than approximately 3.5 um at standard ground-level conditions). The parameters PART_DIAM_MEDIAN and PART_DENSITY are used not only to specify the median particle size diameter and the particle density, respectively, but are also used as flags: if either PART_DIAM_MEDIAN or PART_DENSITY is zero, the ADPIC particles are treated as passive material with no gravitational settling velocity.

The surface deposition flux is modeled separately from gravitational settling. The deposition velocity is typically a function of effluent chemistry, particle size, atmospheric stability, and surface type (e.g. soil, vegetation, etc.). Fig. II.B.6 shows how deposition velocity varies with particle diameter in laboratory and field measurements made by McMahon and Dennison (1979).

The deposition velocity used by ADPIC is linearly weighted according to the distance from ground level, from zero at the height above ground level specified by SFC_DEP_HGT to the full value of SFC_DEP_VEL at ground level. A non-zero deposition velocity will cause an ADPIC particle to be deposited if it is at a height which is less than or equal to the weighted SFC_DEP_VEL (in m/sec) times the time step (in sec). A SFC_DEP_VEL value of zero will result in no removal of ADPIC particles via the deposition velocity mechanism.

If either PART_DIAM_MEDIAN or PART_DENSITY is zero, and if SFC_DEP_VEL are zero, the ADPIC particles are treated as passive material and no dry deposition will result. Although material can be moved into the surface layer from above via advection by the mean wind and by turbulent diffusion, material cannot be moved from the surface layer to the ground through either of these two methods as modeled by ADPIC. Any material that impacts the ground through advection will be reflected.

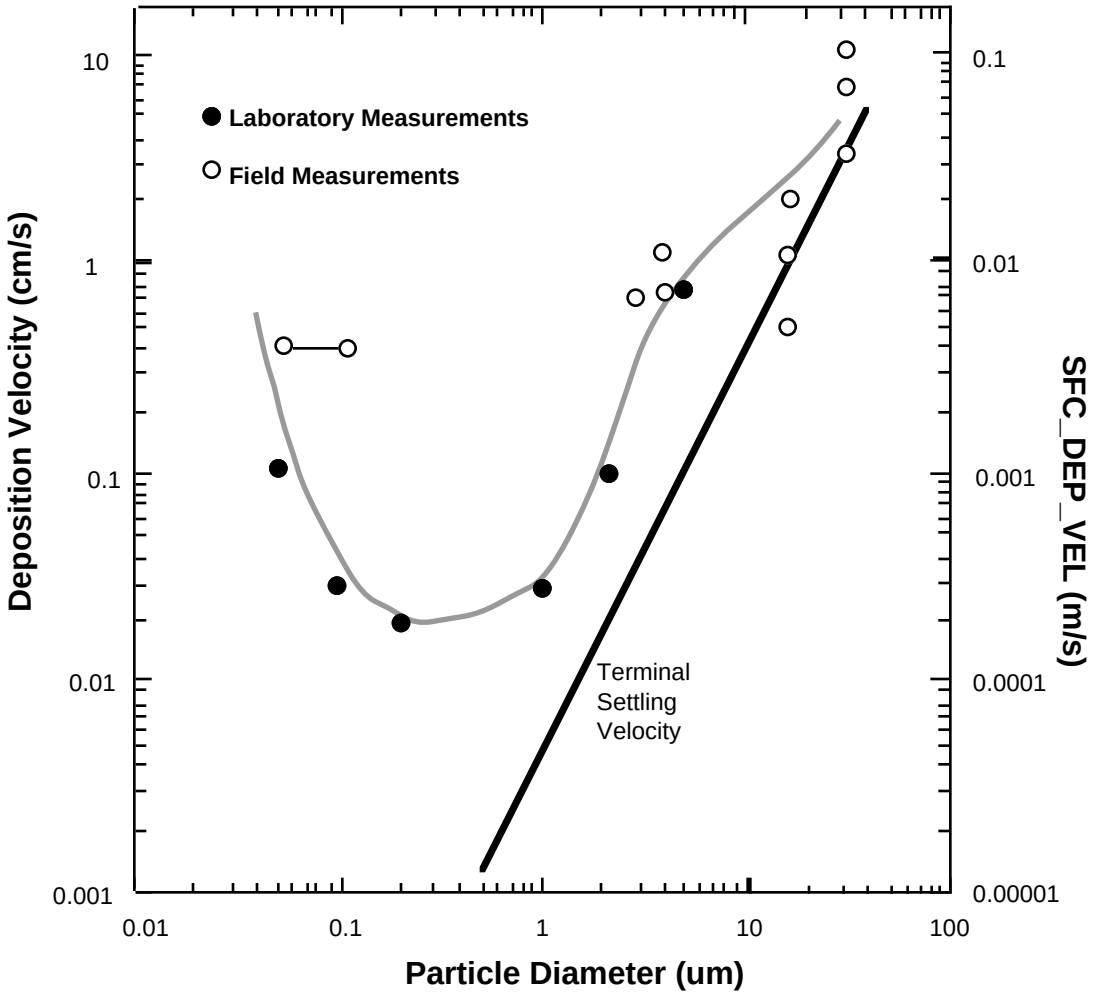


Fig. II.B.6. Laboratory and Field Measurements of Deposition Velocity to Grass vs. Particle Size (McMahon and Dennison, 1979)

II.B.4.e. Washout

At present, the basic assumption for model removal of debris particles in ADPIC is that the base of the rain cloud is above the debris cloud. For surface or near-surface pollutant releases, this seems to be a reasonable assumption to about 100 km downwind. Thus, ADPIC in its present form does not treat in-cloud rainout, but has provisions for below-cloud washout (Crandall et al, 1973).

Washout is treated in the following way in ADPIC. For a specified rain rate over a given surface area, a washout coefficient A (r) is assumed as a function of particles of radius r (see Figs. II.B.7 and II.B.8).

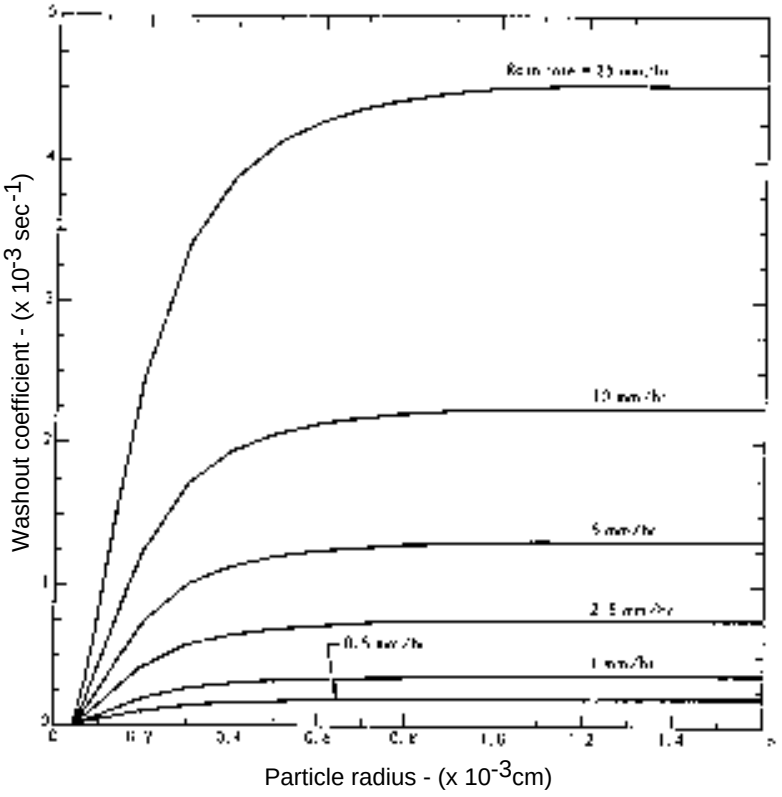


Fig. II.B.7. Washout coefficient versus particle radius for various rain rates. (Crandall *et al.*, 1973)

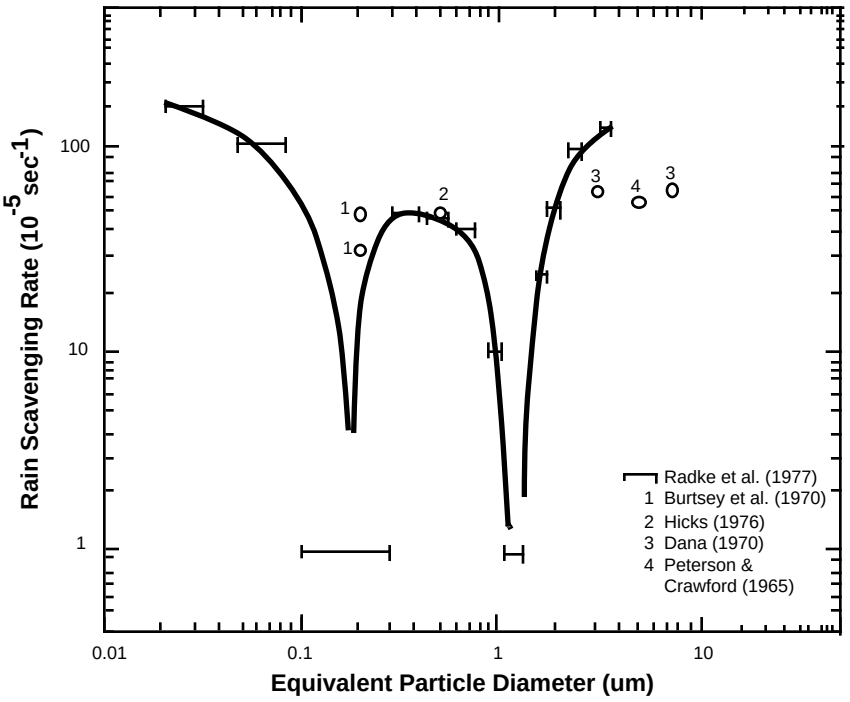


Fig. II.B.8. Measured Values of Scavenging Coefficient vs. Particle Size (McMahon and Dennison, 1979)}

Because of their limited number, each ADPIC particle is only a representation of a certain amount of an actual pollutant. Therefore, an ADPIC mass particle is initially considered to represent an integrated pollutant particle size distribution over sizes such that

$$M(0) = \int_{r=0}^{\infty} m(r,0)\,dr \quad , \tag{II.B.20}$$

where $M(0)$ is the initial mass of an ADPIC particle and $m(r,0)$ is the mass per unit radius of a pollutant particle of radius r at time 0. Constant density is assumed for the pollutant particles. $M(t)$, the mass of an ADPIC particle remaining after a time t in the rain for a given constant rain rate and raindrop size distribution, can be written as

$$\begin{aligned} M(t) &= \int_{r=0}^{\infty} m(r,t)\,dr \\ &= \int_{r=0}^{\infty} m(r,0)e^{-\Lambda(r)t}\,dr \quad . \end{aligned} \tag{II.B.21}$$

To account for the diffusion and transport of pollutants as calculated by ADPIC, this function $M(t)$ is obtained from a table at each ADPIC time step Δt. The fraction of mass $F(\Delta t)$ of an ADPIC particle washed out and deposited on the surface during one ADPIC time cycle then becomes

$$F(\Delta t) = \frac{M(t) - M(t+\Delta t)}{M(t)} = 1 - \frac{\int_{r=0}^{\infty} m(r,0)\varepsilon^{-\Lambda(r)(t-\Delta t)}\,dr}{\int_{r=0}^{\infty} m(r,0)e^{-\Lambda(r)t}\,dr} \quad . \tag{II.B.22}$$

This fraction of mass loss per cycle for each ADPIC particle is stored cumulatively each cycle in a two-dimensional, surface deposition array.

II.B.4.f. Plume Rise

Plume rise can significantly affect ground level concentrations from stack releases into the atmosphere. It becomes important and needs to be evaluated when the exit velocity and/or temperature are sufficiently higher than ambient. Generally, this problem is solved by calculating an effective stack height (physical stack height plus a final plume rise height increment). ADPIC, however, can calculate the time- dependent plume rise of a stack release given the stack emission and meteorological parameters. The parameterizations developed for ADPIC by Desiato and Lange (1986) are used. These parameterizations rely primarily on the work of Briggs (1975) except for the unstable cases where the work of Weil and Hoult (1973) is used. Separate plume rise equations are used for the various combinations of momentum and buoyancy plume rise, vertical and bent-over plumes, and stability class.

The plume rise calculations are controlled with two groups of parameters input via the ADPIC.NML file. The first group are stack parameters for each source: plume rise flag for each source (USE_PLUME_RISE), internal radius of stack (STACK_RADIUS), vertical exit velocity of gas (EXIT_VEL), temperature of gas emitted from stack (STACK_TEMP), and optional sensible heat emission rate (HEAT_EMISSION_RATE). The second group are time-varying meteorological parameters for each source: ambient air temperature (AMBIENT_TEMP), height of strong inversion

affecting plume rise (INV_HGT_AT_STACK) and vertical temperature gradient (VERT_TEMP_GRAD). If STACK_TEMP and AMBIENT_TEMP are specified, then HEAT_EMISSION_RATE should not be specified, and vice versa.

II.B.4.g. Explosive Cloud Rise

An explosive cloud rise module has been incorporated into ADPIC which simulates additional forces acting upon the ADPIC particles due to their proximity to the thermal environment resulting from the detonation of a chemical explosive. When employed, this module generates additional vertical and horizontal components of motion applied to each particle remaining within the calculated effects of the rising thermal cloud.

A full description of the explosive cloud rise integral model code is found in Boughton and Delaurentis (1987), with further discussion of its use in ADPIC discussed in Foster, *et. al.* (1990). Briefly, the code provides a time evolution of the physical and thermodynamic properties of a buoyant cloud formed when a chemical explosive is detonated. The model is based on integrating the three dimensional conservation equations of mass, momentum, and energy over the cloud's cross section. With some simplifying assumptions the integral equations reduce to a set of ordinary differential equations which can be solved for the cloud radius, centerline height, temperature, and velocity as a function of time. These solutions are sensitive to the input parameters describing the ambient atmospheric conditions, especially the atmospheric temperature profile. The initial size and conditions of the cloud are determined using these atmospheric conditions and amount of explosive mass.

Once the cloud's characteristics are determined at each cloud rise model time step (which may be shorter than ADPIC's advection time step), the influence of this buoyant volume must be coupled to the ADPIC particles. There are two considerations in coupling the marker particles to the cloud, the initial loading of the particles into the calculational grid and their subsequent motion while they remain within the buoyant environment. All of the marker particles are initially loaded into the spherical cloud volume at the specified source release time. They have a Gaussian distribution within this volume with radial cutoff values equal to the initial cloud radius as determined by the cloud model. The standard deviations of this distribution are scaled to the cloud radius and are controlled through a scaling factor (SIG_OVER_R used to adjust the “flatness” of the initial distribution. For surface explosions, this initial sphere is tangent to the surface (e.g. it is centered at a height equal to the cloud radius).

As the cloud rises with time it is assumed to contain an environment of enhanced turbulence with entrainment occurring along its surface. Particles remaining inside the sperical cloud volume are treated differently than those moving beyond the cloud radius. Specifically, diffusion is neglected while particles remain within the buoyant cloud. In addition, a fraction of the cloud's vertical motion is applied to each particle within radius, R, of the cloud center. The vertical motion imparted to each

particle due to its proximity to the rising cloud is a function of its distance, r, from the cloud center, and is given by:

$$w = W \exp\left(-c(r/R)^2\right)$$

where

w = the particle's vertical lift due to the rising cloud (in addition to the vertical advection velocity),
W = the cloud's vertical lift motion,
c = a velocity coupling coefficient (= 0.35),
r = the particle's distance from cloud center, and
R = the cloud's radius.

Another difference in the treatment of particles within the cloud is a radially outward adjustment to maintain the particle's relative position between the cloud center and cloud edge as the cloud increases in size. This adjustment “velocity” simulates the effects of a well-mixed cloud, as it grows through entrainment, by redistributing the particles throughout the entire cloud volume. This velocity due to the cloud's expansion is independent from the particle's movement relative to the cloud center due to gravitational fall, the particle's vertical motion w, and potential differences in advection of the cloud and particles due to the spatial variation in the winds.

Therefore, particles remaining within the extent of the cloud are moved by applying the appropriate advection, expansion, fall, and vertical lift velocities. Once a particle leaves the influence of the rising thermal (usually due to the particle's gravitational fall velocity moving the particle beyond the cloud radius), only advection, diffusion, and gravitational settling processes act on the particles.

The dynamics of the cloud are followed until its vertical velocity decreases below a predetermined limit. At this point all particles are treated as though the cloud no longer exists and the cloud calculation is bypassed for the remainder of the problem. A similar effect occurs if the cloud center rises above the top of the computational grid.

The use of the explosive cloud rise module is controlled by the ADPIC.NML parameter USE_EXPL_CLD_RISE; the remaining explosive cloud rise parameters are contained in the ADPIC_EXPL_CLD namelist in the ADPIC.NML file. Although multiple ADPIC sources may be put into a single explosive cloud (controlled by parameter SRC_EXPL_CLD), the capability for multiple explosive clouds does not yet exist. The model used in ADPIC assumes a high explosive device (HE). Other explosive cloud source parameters include heat of detonation (HEATDET), high explosive amount (HEAMT), velocity coupling coefficient (VEL_COUPL), and height of the center of the explosive cloud (ZINIT). Meteorological parameters required for the explosive cloud rise module include relative humidity at the surface (REL_HUM), atmospheric pressure at the source center height (SRC_PRESS), and the upper air temperature profile (TEMP_PROFILE). The content of the output text file is controlled by the parameters PRINT_EXPL_CLD and PRINT_TYPE_EXPL_CLD.

II.B.4.h. Hybrid Particles *

In the hybrid-particle model all the constituents of a pollutant that have similar transport properties can be treated as a single source. The number of sources typically carried can still be up to nine; however the number of parent nuclides followed can be up to 100. In many problems, one source will be sufficient. Although designed for radioactive sources, the hybrid model will properly handle non-radioactive sources, in which case release rates would be specified in units of mass. As in the simple-particle model, each source is represented by a collection of marker particles, typically a few thousand. Each marker particle represents the set of nuclides in the source, hence the name hybrid particle. As already stated, all the nuclides of the set must have similar transport and diffusion properties. Some examples of such sets are noble gasses, elemental halogens, refractory oxides, alkali metals, and noble metals.

The time dependence of the activity and toxicity for each source of hybrid particles is calculated in tables at the beginning of the problem and applied at times of interest during the calculation. The presence of radioactive decay products are included in the calculations. This requires the solution of a system of coupled ordinary differential rate equations, whose details are described below. Figure II.B.9 illustrates a schematic representation of the structure described above.

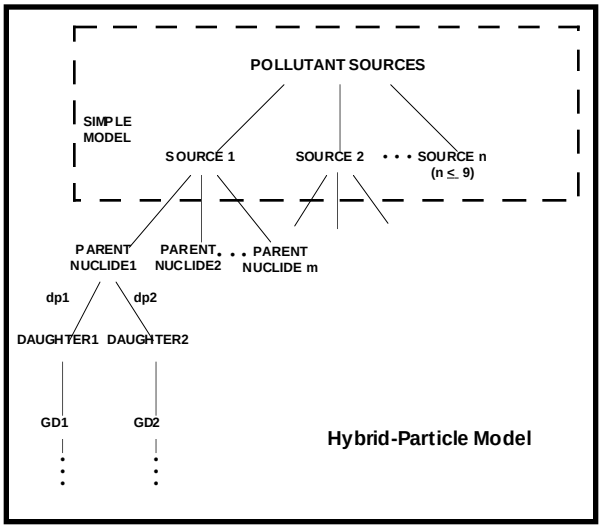


Fig. II.B.9. A schematic representation of the source structure. The dashed box defines the limits of the simple model. The complete box defines the hybrid-particle model. Only the first branch of the hybrid is depicted here. Parent nuclide 1 decays by decay path 1 to daughter 1 and by decay path 2 to daughter 2. Daughter 1 decays to granddaughter 1 and so on. More discussion on the chain decay process is given below.

---

* This section was taken from "A Description of the ADPIC Hybrid-Particle Model", Freis and Harvey, UASG95-4

HP Activity and Toxicity

Since each marker particle in a hybrid source can represent several nuclides, the set of initial nuclides in a source and their amounts must be determined at $t = 0$ from problem input. Changing this set of initial nuclides requires a new source. For a given marker particle the ratio of nuclides changes with time, depending on the nuclide half-lives and branching ratios. Chain-decay products add additional nuclides into the hybrid particle with time. The total hybrid-particle activity is the sum of the activity carried by each of the nuclides. We discuss this in detail here.

The activity for a single radionuclide is given by the expression

$$\dot{N}(t) = \lambda \; N(t) \qquad \text{(II.B.23)}$$

where N(t) is the number of atoms at time t and λ is $ln\ 2$ divided by the radionuclide half life. The next section discusses how N(t) is determined for all the nuclides in a decay chain starting with a single parent. In this section we assume this is known and use that information to determine the activity and toxicity.

The total activity at time t from one source released at time t_R is

$$C(t,t_R) = \sum_{j=1}^{jmax} \sum_{i=1}^{imax(j)} \lambda_{ij} N_{ij}(t - t_R) \qquad \text{(II.B.24)}$$

where we have summed over *jmax* parents and *imax(j)* daughters (note: each *imax* depends on *j*.)

We introduce the time dependent release rate; $\dot{R}(t)\ A_0$ where $\dot{R}(t)$ is normalized such that $\int_0^\infty \dot{R}(t)\ dt = 1$ and A_0 is the total activity released over all time by this source. Then the activity released in dt at $t = t_R$ is

$$C(t_R,t_R) = \dot{R}(t_R)\ A_0\ dt \qquad \text{(II.B.25)}$$

In the Hybrid-Particle Model we normalize each parent nuclide so that its initial decay rate is one curie. Thus, suppressing subscripts, the normalized activity in curies of the parent nuclide is defined by

$$\tilde{N}(0)\ \lambda = 1$$

and

$$N(t) = \kappa\ \tilde{N}(t) \qquad \text{(II.B.26)}$$

where κ is a proportionality constant to be determined .

The relative amount of each parent nuclide present at t = t_R is specified as F_j where

$$\sum_{j=1}^{jmax} F_j = 1$$

Then, in terrms of the normalized variables, the total activity at time *t* from one source realeased at time t_R is

$$C(t,t_R) = \sum^{jmax} F_j \sum^{imax(j)} \lambda_{i,j}\, \kappa\, \widetilde{N}_{i,j}(t - t_R)$$

(II.B.27)

To determine κ, evaluate $C(t,t_R)$ at t = t_R, when no daughters have yet been produced. Then,

$$C(t_R,t_R) = \sum_{j=1}^{jmax} F_j\, \lambda_{1j}\, \kappa\, \widetilde{N}_{1j}(0)$$

and since

$$\sum_{j=1}^{jmax} F_j\, \lambda_{1j}\, \widetilde{N}_{1j}(0) = 1$$

then

$$C(t_R,t_R) = \kappa \quad .$$

From Eq. 25

$$\kappa = \dot{R}(t_R)\; A_0\; dt$$

The total activity at time *t* from one source realeased at time t_R (Eq. 27) becomes

$$C(t,t_R) = \dot{R}(t_R)\, A_0\, dt \sum^{jmax} F_j \sum^{imax(j)} \lambda_{ij}\, \widetilde{N}_{i,j}(t - t_R)$$

(II.B.28)

The activity per marker particle is calculated by dividing Eq. 28 by the number of marker particles released during dt at t = t_R,

$N_p(t_R) = \dot{N}_p(t_R)\; dt$ where $\dot{N}_p(t_R)$ is the release rate of marker particles.

Then $C_p(t,t_R)$, the activity per marker particle is

$$C_p(t,t_R) = \frac{\dot{R}(t_R)\, A_0}{\dot{N}_p(t_R)} \sum^{jmax} F_j \sum^{imax(j)} \lambda_{ij}\, \widetilde{N}_{ij}(t - t_R)$$

(II.B.29)

The toxicity (dose rate) for each nuclide is calculated by multiplying it's activity by an appropriate dose conversion factor which takes into account biological factors such as the organ exposed, the exposure pathway (e.g. inhalation, cloudshine) and exposure time. For a single radionuclide the dose rate is given by;

$$\dot{D}(t) = DCF_k \ \lambda \ N(t) \qquad \text{(II.B.30)}$$

where DCF_k is the appropriate dose conversion factor for the toxic effect *k*, for a particular pathway. Using the same steps as was done above for the activity, and replacing Eq. 23 with Eq. 30, the toxicity per marker particle, $T_{p,k}(t,t_R)$, at time t from particles released at t = tR is

$$T_{p,k}(t,t_R) = \frac{\dot{R}(t_R)\, A_0}{\dot{N}_p(t_R)} \sum^{jmax} F_j \sum^{imax(j)} DCF_k \ \lambda_{ij} \ \tilde{N}_{ij}(t - t_R) \qquad \text{(II.B.31)}$$

Having determined the activity/toxicity per marker particle the calculation of distributing it over the Eulerian sampling grid is the same as in the simple model (i.e., bilinear interpolation to the neighboring grid points for surface deposition and trilinear interpolation to the nodes of the surrounding volume element for airborne samples)..The units of the DCF are $\left[\frac{rem\ m^3}{s\ Ci}\right]$ for airborne samples and $\left[\frac{rem\ m^2}{s\ Ci}\right]$ for deposited surface samples. The total dose rate from a fixed postiion on the Eulerian grid is the sum of the contributions from the marker particles. The total dose from a fixed position is the time integral of the dose rate over the time interval of exposure.

Nuclide Time Dependence Equations

Both the activity from Eq. 29 and the toxicity from Eq. 31 depend upon $N_{i,j}(t)$, the amount of each nuclide present at time t.* $N_{i,j}(t)$, is determined by solving a separate set of coupled ordinary differential rate equations for each nuclide initially present. These equations describe the parent-daughter chain decay process and require the branching ratio and half life for each nuclide in the chain. For all nuclides of practical interest this process has several special features :

At t = 0, only the parents are non-zero (i.e., $N_{ij}(0) = 0$ for all $i > 1$)

Each generation has at most two members. In many cases (but not all) the two members are the ground state and meta-stable state of the same element. In such cases there is a one to one correspondence between elements and generations.

The parent for any nuclide is in the immediately preceding generation or, in the case of meta-stable to ground transitions, is in the same generation.

---

* (In this section we supress the ~ but we are working with the normalized quantities. Also the release time, t_R , is in this section the initial time t = 0.)

The half life for each member is unique (i.e., no two memebers of the family have the identical half life to all significant figures. This feature is important for the solution method employed) .

The maximum number of generations is small (i.e., less than 20).

Utilizing these features, the set of equations has the following form in which each nuclide is uniquely identified by two indices, the first being its position within the generation (1 or 2), and the second being its generation number.

$$\dot{N}_{1,1} = -\lambda_{1,1} N_{1,1}$$
$$\dot{N}_{1,2} = a_{1,1,2}\lambda_{1,1} N_{1,1} - \lambda_{1,2} N_{1,2}$$
$$\dot{N}_{2,2} = a_{2,1,2}\lambda_{1,1} N_{1,1} + b_{2}\lambda_{1,2} N_{1,2} - \lambda_{2,2} N_{2,2} \qquad \text{(II.B.32)}$$
$$\dot{N}_{1,3} = a_{1,1,3}\lambda_{1,2} N_{1,2} + a_{1,2,3}\lambda_{2,2} N_{2,2} - \lambda_{1,3} N_{1,3}$$
$$\dot{N}_{2,3} = a_{2,1,3}\lambda_{1,2} N_{1,2} + a_{2,2,3}\lambda_{2,2} N_{2,2} + b_{3}\lambda_{1,3} N_{1,3} - \lambda_{2,3} N_{2,3}$$

$$\dot{N}_{i,g} = \sum^{2} a_{i,j,g}\lambda_{j,g-1} N_{j,g-1} + (i-1) b_{g}\lambda_{1,g} N_{1,g} - \lambda_{i,g} N_{i,g} \qquad \text{(II.B.33)}$$

where

i = nuclide position

g= nuclide generation

a, b = decay branching ratios. b is for gama decays to ground state (intragenerational), and a is for β decays to another element (intergenerational).

For convenience all meta-stable nuclides have position 1, hence nuclide $N_{1,g}$ can be a parent to nuclide $N_{2,g}$ in a given generation but the reverse is not allowed.

The Equation Solution

The analytic solution of the set of equations represented by Eq. II.B.33 is (See Appendix E for derivation)

For the first generation (g = 1)

$$N_{1,1}(t) = (1/\lambda_{1,1})\, e^{-\lambda_{1,1} t}, \qquad A(1,1,1) = 1/\lambda_{1,1}$$

and for subsequent generations (g > 1)

$$N_{i,g}(t) = \sum_{l=1}^{L_{max}-1} A(l,i,g) e^{-\lambda_{1,1} t}, \; g > 1$$

(note: $A(l,i,g)$ is defined below)

where

$$L_{max} = 2(g-1) + i - 1$$

$$\bar{\lambda}_1 = \lambda_{1,1}$$

$$\bar{\lambda}_l = \lambda_{\hat{i},\hat{g}}, \qquad l = 2(\hat{g}-1) + \hat{i} - 1, \quad \text{for } g \geq \hat{g} > 1 \qquad \text{(II.B.34)}$$

The $(\hat{i}, \hat{g})$ pair related to l is determined from Eq. II.B.34. Note $\hat{i}$ = 1 or 2 .

The coefficients $A(l,i,g)$ are solved for recursively as follows:

for $l = 1. \ldots . L_{max} - 1$

$$A(l,i,g) = \sum_{j=1}^{2} \left[a_{i,j,g}\, \lambda_{j,g-1} A(l,j,g-1) + (i-1)\, b_g\, \lambda_{1,g} A(l,1,g) \right] \left(\bar{\lambda}_{L\,max} - \bar{\lambda}_l \right)^{-1},$$

and

$$A(L_{max},i,g) = -\sum_{l=1}^{L_{max}-1} A(l,i,g)$$

Ground Deposition Accumulation

Particles reaching the ground are saved in a list that includes their age, deposition time, and deposition position for processing as needed. At sampling times they are assigned their appropriate activity and toxicity and distributed in sampling bins defined on an Eulerian grid. This permits

accurate decay in sampling bins that may contain multiple sources. This also enables total time integrated deposition dose to be calculated.

Rainout deposition, which removes a fraction of each active particle each time step, is treated differently since a deposited particle list would quickly grow to an unmanageable length. To avoid this the rainout particles are saved in Eulerian sampling bins with an additional index for particle age (or born time). This permits the sampling bins to be accurately decayed, but limits the bin to a single source. At deposition sampling times the rainout bins are added to the regular deposition bins, however they cannot be included in the total time integrated deposition calculation.

Implementation Comments

The use of the ADPIC_RADPARAMS module is controlled by the ADPIC.NML parameter DECAY_MODE being set to 'rad_decay'. The Hybrid source is created using NUCLIDE_NAME to specify the nuclides in a single source and NUCLIDE_MIX to specify the percentage that nuclide contributes to the total. The pathway a source uses is set using DOSE_PATH_DESCRIPTOR. From the NUCLIDE_NAME and DOSE_PATH_DESCRIPTOR, comes the DCF_NUCLIDE_NAME (the name of the parent or daughter nuclide), DCF_PATH_DESCRIPTOR (same as the DOSE_PATH_DESCRIPTOR but for each DCF_NUCLIDE_NAME), and the DCF (the dose conversion factor for each nuclide and it's specified pathway). Hybrid ADPIC applies the proper dose conversion factors to the parent nuclide and its daughters for the pathways specified for each source. Other required parameters are: NUMBER_OF_TOXIC_TIMES (the number of entries in the hybrid_toxicity_array), and DECAY_START_INTERVAL (which allows you to set when you wish the decay process to begin).

II.B.4.i. Moving Receptors

Background

As part of a Bio Sensor Architecture Study, a "moving receptor" capability has been implemented in ADPIC. This capability allows users to move one or more receptors (samplers) through the ADPIC domain and accumulate a quantitative integrated exposure from the ADPIC cloud to each of these receptor(s). Note that only an integrated and interval averaged exposure from the airborne cloud is computed. The capability to determine the time-dependent instantaneous air exposure or exposure from deposited material is not currently implemented. The implementation required for these other types of exposure is significantly different from that needed for integrated air.

The current implementation is based on the user specifying one or more 3-dimensional (x,y,z) tracks (trajectories) through the ADPIC domain as a function of time. One or more receptors are then associated with a particular track. If more than one receptor is associated with a track, then the user may also need to specify the receptor's x, y, z offset from the track. This allows a 1-, 2-, or 3-dimensional grid of receptors to move in unison throughout the domain. Possible uses of this

capability include being able to determine the exposure received by an airplane flying on a particular course within the model grid, or a vehicle traveling a specified course within the model grid, or a person walking along a specified course within the model grid.

Note that having a few receptors for an ADPIC run does not seriously add to the wall-clock-time require for the model run; however, a few tens (or more) of receptors will significantly slow the execution time.

Time step control in ADPIC when using the moving receptors option is applied at two levels. First, global time step is altered downward if necessary so that a rcptr_interval time is not overshot. Second, the local time step used in moving individual particles is constrained so that the fastest moving receptor will not move more than one cell's distance in the horizontal or the vertical during that time step. For the case of rcptr_hgt_ref = 'terrain", this means that the implied vertical velocity including the velocity due to traversing terrain blocks is used. Also, the most constraining horizontal sampling grid dimension (including nested grids) at the particle's location is used in the horizontal.

Assigning Receptors to Sampling Bins

The user may specify identifier strings to the receptors via the RCPTR_ID parameter. To specify the combination of ADPIC sources to be sampled by the receptors, use the RCPTR_SBIN parameter to match each receptor with the desired sampling bin. For example:

SOURCE_TO_SAMPLING_BIN(1,1) = 1 2 3

SOURCE_TO_SAMPLING_BIN(1,2) = 2 3 4

SOURCE_TO_SAMPLING_BIN(1,3) = 1

.

.

.

RCPTR_ID(1) = 'one' 'two' 'three'

RCPTR_SBIN(1) = 2 1 3

This assigns receptor 'one' to integrate air concentrations from sampling bin 2 (i.e., ADPIC sources 2, 3, and 4). Sampler 'two' integrates sampling bin 1 (i.e., ADPIC sources 1, 2, and 3. Sampler 'three' integrates sampling bin 3 (i.e., only the concentrations from ADPIC source 1). NOTE: the SAMPLING_TYPE of the sources assigned to the receptors must not be 'INTDEP' or 'TOTDEP'. (Even though any "air" concentration SAMPLING_TYPE is okay, remember only integrated and interval averaged concentrations are calculated for the receptors).

Receptor Height Reference

The z coordinates used for the receptor trajectories may be referenced (via the RCPTR_HGT_REF parameter) as height above terrain, height above sea level, or height above grid

bottom. Locations specified in the TRAJ_X/Y/Z arrays are checked to make sure that none are below the surface. The user should also be especially careful in specifying the trajectories (and offsets) when using a RCPTR_HGT_REF set to other than 'TERRAIN', since it will not be obvious by just looking at the model namelist input file that the receptor may pass through terrain when going from one set of coordinates to another. If the receptor passes through the terrain at any time, the user will be warned, and the receptor will cease to accumulate exposure until such time as it reappears above the surface.

Defining Base Trajectories, Optional Offsets, and Assigning to Each Receptor

The time-varying trajectory coordinates are specified using the TRAJ_X, TRAJ_Y, and TRAJ_Z arrays, with TRAJ_TIME and TRAJ_DATE defining the times for which each of the sets of coordinates is valid. Specify all the dates and times at which any of the trajectories need to have their beginning/intermediate/final locations set. Trajectories move in a linear fashion between these locations (except when RCPTR_HGT_REF = 'TERRAIN', the movement in the z-direction follows the terrain) at a rate dictated by the associated dates/times. For example (in this case only trajectory numbers 2 and 4 are being used by the current set of input):

TRAJ_DATE	=	6*'01JAN96'					
TRAJ_TIME	=	'01:00'	'02:00'	'02:05'	'02:30'	'03:00'	'04:00'
TRAJ_X(1,2)	=	295.2	300.1	300.1	296.2	290.5	-99999.0
TRAJ_X(1,4)	=	-99999.0	-99999.0	-99999.0	325.9	330.2	-99999.0
TRAJ_Y(1,2)	=	4201.0	4206.2	4206.2	4220.2	4221.0	-99999.0
TRAJ_Y(1,4)	=	-99999.0	-99999.0	-99999.0	4150.3	4150.5	-99999.0
TRAJ_Z(1,2)	=	50.0	-99999.0	-99999.0	-99999.0	-99999.0	40.0
TRAJ_Z(1,4)	=	-99999.0	-99999.0	-99999.0	20.0	-99999.0	-99999.0

The RCPTR_TRAJ parameter is used to associate receptor(s) to particular trajectories. For example:

RCPTR_TRAJ = 2 2 4

assigns receptors 'one' and 'two' to follow trajectory number 2, and receptor 'three' to follow trajectory number 4.

The user may also define receptor offsets from their associated trajectory. For example, if you wanted receptors 'one' and 'two' to move together but maintain a constant 5 km offset in the x direction, a 10 km offset in the y direction, and a 15 m offset in the z direction, and if you also wanted the two receptors to be symmetrically centered on their associated trajectory, you would specify:

```
RCPTR_DELX    =  -2.5    2.5

RCPTR_DELY    =  -5.0    5.0

RCPTR_DELZ    =  -7.5    7.5
```

Since we wanted receptor 'three' to follow the originally specified trajectory, you can just leave out the 3rd position of the RCPTR_DELX/Y/Z arrays, since the default is 0.0.

In summary, for the example trajectories (with the example offset specifications) and associated receptor assignments, the following will happen:

1) Receptor 'one' will move

   from location x=292.7 (295.2-2.5), y=4196.0 (4201.0-5.0), z=42.5 (50.0-7.5) at 01:00

   to location x=297.6, y=4201.2, z=39.167((50-7.5) - ((50-40)/(4-1))) at 02:00.

2) Receptor 'two' will move in a similar manner, only offset in the x, y, and z directions as indicated earlier.
3) Between 02:00 and 02:05 both 'one' and 'two' will remain stationary in the x and y dimensions, but will continue movement downward. They will then begin to move in all three dimensions once again at 02:05.
4) Receptor 'three' is relatively simple, moving only between 02:30 and 03:00 and remaining at a constant elevation of 20 m above terrain.

Note that receptors will not accumulate any exposure prior to the first time, or after the last time, a valid location coordinate is listed for its associated trajectory (in either the x, y, or z direction). In this example, 'one' and 'two' accumulate exposure from 01:00 up until 04:00. Receptor 'three' only "exists" and accumulates exposure between 02:30 and 03:00. In other words, a receptor is removed from the calculations as soon as it has no where else to go during the remainder of the run.

Range of Influence of Particles to Receptors

A receptor's integrated exposure concentration increases whenever an ADPIC particle moves within a user-controllable range of the receptor. A test is made after each local time step for each particle to see if the particle is within the specified range of any receptor. If so, it contributes a portion of its mass/activity to the receptor based on the "tent-function" method normally used for the sampling process to grid points . The range which is used for the particle-to-receptor proximity test

is controlled by RCPTR_INFL_X_FRACT, RCPTR_INFL_Y_FRACT, and RCPTR_INFL_Z_FRACT. The values (defaults = 1) assigned to these parameters are the fractions of the extent of the advection grid cell over which the tent function reaches, in the respective directions. For example:

RCPTR_INFL_X_FRACT = 0.5

RCPTR_INFL_Y_FRACT = 0.25

would create a sampling reach of 1 delta Z, 0.5 delta X, and 0.25 delta Y.

Receptor Concentration Multipliers

For purposes of converting ADPIC concentration units to desired output units when using the moving receptors option, three hierarchical input parameters can be used as multiplier values. RCPTR_MULT_GLOBAL is used to assign a single multiplier to every receptor output value; RCPTR_MULT_GROUP is used to assign a multiplier to one or more "groups" of receptors (a "group" of receptors are those associated with a particular trajectory); RCPTR_MULT_IND is used to assign a multiplier to one or more specific receptors. RCPTR_MULT_GROUP assignments override RCPTR_MULT_GLOBAL assignments, and RCPTR_MULT_IND assignments override both RCPTR_MULT_GLOBAL and RCPTR_MULT_GROUP. For example:

RCPTR_MULT_GLOBAL	=	2.5		
RCPTR_MULT_GROUP	=	-99999.9	4.2	(index is trajectory number)
RCPTR_MULT_IND	=	-99999.9	3.0	(index is receptor number)

will assign a global multiplier to all three receptors of 2.5, then this value is overridden for those associated with trajectory number 2 (receptors 'one' and 'two', as specified by the RCPTR_TRAJ parameter) by assigning a multiplier of 4.2, and finally receptor 'two' again is reassigned a different value of 3.0. So in this example the integrated exposure concentrations of receptor 'one' are multiplied by 4.2, 'two' are multiplied by 3.0, and 'three' are multiplied by 2.5.

Output

The frequency of output of the receptor concentrations is controlled by the RCPTR_INTERVAL parameter. All output generated during ADPIC is stored in memory and output at the end of the model run into a file named RECEPTOR.DAT in the dispersion run directory. For example:

RCPTR_INTERVAL = '00::00:10'

will output integrated exposure (multiplied by any stated multiplier values specified by RCPTR_MULT_GLOBAL, RCPTR_MULT_GROUP, and RCPTR_MULT_IND) for each receptor every 10 minutes of model run time. Note that the list of concentrations for a particular receptor

does not start until a non-zero concentration is found for the receptor, and the list stops when all remaining integrated values are constant.

II.B.4j. Decay Mode

Through the use of the DECAY_MODE parameter in the ADPIC_SOURCE namelist, the user is able to specify if and how the source material will decrease in activity or amount, or change its form, state, or other characteristics. The decay method used for the hybrid particles option is discussed in more detail in section II.B.4.h.

Simple radioactive decay and chemical or biological decay are simpler decay modes discussed in this section. The following equation specifies the decrease in time (t) of the amount or activity of material (N) using a decay constant, .

$$\frac{dN}{dt} = -\lambda t \qquad \text{(II.B.35)}$$

which can also be expressed as:

$$\lambda = \frac{-\ln\left(\frac{N}{No}\right)}{t} \qquad \text{(II.B.36)}$$

For simple radioactive decay, the decay constant is expressed in terms of a half-life ($T_{1/2}$), which is the time it takes for the material to decay to half of its original amount (No):

at t= $T_{1/2}$, $N = \frac{No}{2}$

Substituting into Eq. II.B.36 :

$$\lambda = \frac{\ln 2}{T_{1/2}} \qquad \text{(II.B.37)}$$

For chemical or biological decomposition, the decay constant is expressed in terms of τ , which is the time it takes for the material to decompose to 1/e of the initial amount:

at t= τ, $N = \frac{No}{e}$

substituting into Eq. II.B.36:

$$\lambda = \frac{1}{\tau} \qquad \text{(II.B.38)}$$

II.B.4.k. Meteorological Parameters

Several time-varying meteorological or meteorology-related parameters that affect the diffusion and/or deposition of pollutant sources in ADPIC can be input via the ADPIC.NML namelist file. These time-varying met parameters (TURB_PARAM_TYPE, STAB_CLASS, INV_OBUKHOV_LEN_ML, INV_OBUKHOV_LEN_SL, ML_HGT, SIGMA_THETA_FACT, and PRECIP_SCAVENG_COEF) may have up to 50 values, with the MET_START_DATE and MET_START_TIME parameters controlling the start date and time that each value will become effective.

II.B.4.l. Nested Grid Sampling

A system of nested concentration sampling grids has been devised to improve the near-source resolution of the concentration values calculated from particle positions. The largest grid in Fig. II.B.10. represents ADPIC's standard advection and outermost concentration sampling grids, (i.e., the R41 grid, see Appendix F) dimensioned 40 x 40 cells. All interior, nested grids represent concentration sampling grids of diminishing size, each having half the number of cells as the main grid, i.e., 20x20 in the figure, at half the resolution of the next outer grid. In all cases, the vertical dimensions of the advection and sampling cells are identical. The source maintains the same relative position in each of the five grids. To demonstrate how these grids are situated in relation to one another, assume that the Model x coordinate and northing coordinates of the source are 410.5 km and 4011.3 km, respectively, and that the assessor selects an advection grid for ADPIC whose lower-left coordinates are 400 km and 4000 km in the easting and northing directions. (Skewing the grid in this fashion would be appropriate when the prevailing winds and the winds in the foreseeable future are from the southwest.) Assuming a standard ADPIC grid having 40 x 40 cells in the horizontal with Δx and Δy equal to 2 km, the source would then occupy the advection cell whose index location is $i = 6$ and $j = 6$. Since the nested grids have 20 x 20 cells, the source should reside on or within the $i = 6$, $j = 6$ cell of all nested grids. (The word “should” is used here since a nested grid, if necessary, is shifted to the east and south so that its boundary falls on the nearest mesh lines of the next largest grid.) After receiving this coordinate information from the TOPOG*.GRID file (see Section III.C.1.), ADPIC correctly positions each of the four nested grids.

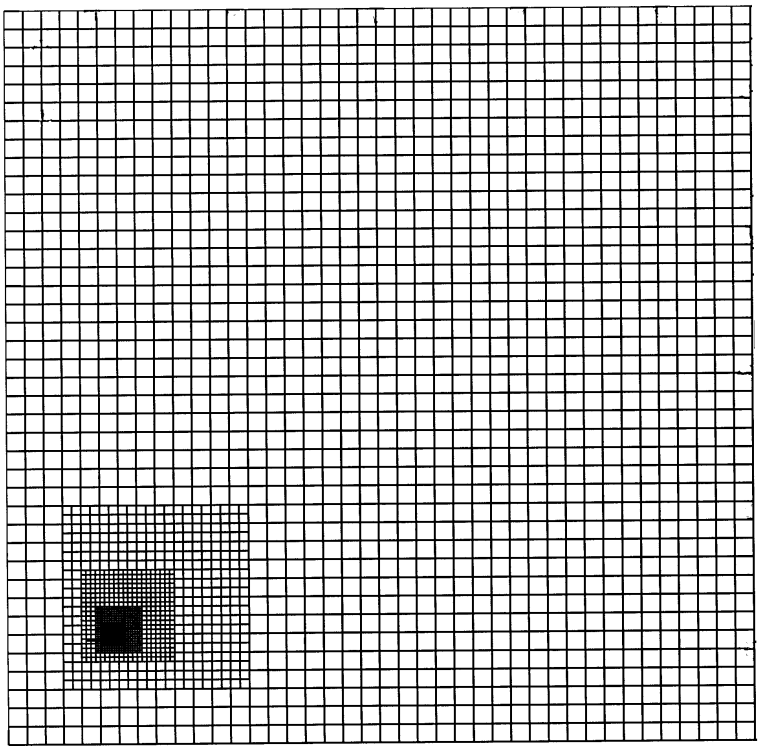


Fig. II.B.10. Nested grids for improving near-source sampling

ADPIC produces a set of five two-dimensional arrays of concentration at each of 1 to 10 vertical levels. These arrays are used by the plotting code, PLCNT, to generate isopleths of concentrations. It should be clear from the previous discussion that several arrays can share a common space. When this occurs, PLCNT will always choose, in a spatial sense, the more highly resolved values of concentration. Use of the nested grid approach does, however, create complications; immediately beyond the extremity of any nested grid, the cell volume increases by a factor of four. This causes discontinuities in a contour when it spans two or more nested grids because of an abrupt drop in concentration. To improve the presentation, a routine has been developed that smooths the concentrations at the interface of two nested grids. However, use of this routine places a constraint on the ADPIC grid definition: the source must never be within two cells of the advection grid boundary. Returning to our example, a UTM source location of 402 km east and 4010 km north would violate the two-cell condition, assuming that all grid selections remained the same.

There are four nested grids that are used along with each of these arrays. Each of the nested grids has a Δx and Δy one-half that of its next larger neighbor. These nested grids are only used for sampling concentrations. The standard, outermost computational grid is the same as that for the input gridded wind fields, and is the only one used for the advection calculation. These nested grids are fixed in that there are always four of them and their cell size is determined by the size of the computational cells.

The placement of the nested grid is determined by the placement of the source in the computational grid. The nested grid is automatically placed such that the source is in the same relation to the nested grid as the source has with the computational grid. The only restriction is that when the grid is originally set up, the source must be at least two grid cells away from the computational boundary. If the source has been placed too close to the outside, a warning message will be printed out in ADPIC with the option to continue or start over.

These concentration arrays are used by PLCNT to plot contours of the concentrations. In addition to these concentrations, ADPIC can also print tabular output of such things as particle coordinates, advection velocities, concentrations, and friction velocities.

II.B.4.m. Output

The contents of up to MAXSBINS (currently 30) ADPIC output bins may be specified using the parameters SOURCES_TO_SAMPLING_BIN, SAMPLING_TYPE and SAMPLING_HGT. Any particular ADPIC source may be entered more than once. Also, combinations of ADPIC source numbers may be entered to sum the contributions from more than one ADPIC source.

II.B.5. Precision of ADPIC Calculations

Inasmuch as the ADPIC particles represent statistical sample of fluid elements marked with species mass, it is desirable to have as many particles per concentration sampling cell as possible. The upper limit of this number is obviously dictated by computer storage. ADPIC assigns to each particle a fictitious cell volume of the same size and shape as the basic concentration sampling grid cell. Thereby, in calculating the concentration field from the particles, each particle will contribute some fractional part of its mass to eight nearest-neighbor cells in proportion to the overlap of its “volume” with the cell volumes of the neighbor cells. This procedure smooths out the concentration field. From many prior calculations, we find that, due to the smoothing technique as few as one particle per cell on the average will yield meaningful results. The statistical significance of the calculated concentrations is obviously dependent on particle count; the fractional error of the concentration is inversely proportional to the square root of the particle count.

II.C. Support Routines

II.C1. TOPOG

The construction of a solid bottom boundary for the CG-MATHEW and ADPIC models is accomplished through TOPOG. The boundary can be viewed as a system of blocks which approximate the shape of the actual terrain. Therefore the results of subsequent CG-MATHEW and ADPIC runs will reflect the major topographic features in the area of interest. The system of blocks is derived by averaging data from a problem elevation data base (see Section III.B.3.) to a grid volume defined by the user (see Section III.B.1.). Because of the large amount of averaging that has been applied to the original digitized terrain data in the generation of the system of blocks, the model terrain produced automatically by TOPOG may not be the best representation of terrain possible for a given grid resolution. ARAC's in-house version of TOPOG allows a user to selectively and interactively raise or lower individual boundary cells so as to more closely conform to the actual surface. Appendix C provides a description of this interactive capability.

The definitions of the CG-MATHEW and ADPIC grid volumes produced by TOPOG are made up of grid parameters and terrain heights for each horizontal grid cell. The user selects a grid by assigning values to certain grid parameters in the TOPOG.NML file (see Section III.B.1.). The ADPIC and CG-MATHEW grids are subject to a number of constraints in order to ensure proper averaging. TOPOG checks the grid parameters and if they define an illegal grid, TOPOG either halts with diagnostic errors or makes minor adjustments to some of the parameters in an attempt to create a legitimate grid.

In a typical ARAC problem, the user specifies the center of the ADPIC grid in the x and y coordinates of the map projection selected for the problem (optionally the user can specify the coordinates of the lower left corner of the ADPIC grid). Currently, most regional problems are run on the Universal Transverse Mercator (UTM) projection. The user also specifies the x, y and z ranges of the grid (or optionally the x, y, and z dimensions of an individual cell). Since the number of grid points in each dimension is normally left constant, choosing the cell size is equivalent to specifying the range of the ADPIC grid (e.g., an ADPIC grid having 41 by 41 by 15 points with a cell size of 1 km by 1 km by 50 m defines a grid volume of 40 km by 40 km by 700 m). The CG-MATHEW grid is defined on the basis of the user's ADPIC grid. They have the same grid cell sizes, but the number of points in CG-MATHEW's grid system (in any direction) must, at a minimum, be as great as the number in ADPIC's grid system. The CG-MATHEW grid surrounds the ADPIC grid so that a band of cells borders on all sides of the ADPIC grid. This has a buffering effect that protects ADPIC from possibly poor adjustments of the winds near CG-MATHEW's boundaries. For the standard ARAC model grid dimensions (CG-MATHEW: 51 by 51 by 15 grid points, ADPIC: 41 by 41 by 15 grid points, i.e., the R41 grid, see Appendix F), the band surrounding the ADPIC grid is usually five grid cells wide.

The terrain averaging requires that the boundaries and origins of both the CG-MATHEW and ADPIC grids lie on top of grid points in the elevation data base grid The current averaging procedure also requires that both of the horizontal grid cell lengths (DELX, DELY) be integer multiples of one-quarter the resolution of the terrain data base. Since standard regional elevation grids have grid cells which are 500m by 500m, legitimate values for DELX and DELY are integer multiples of 125 m. For hemispheric problems the grid cells must be 381 km by 381 km. The constraints on the cell size are absolute. TOPOG will halt if illegal values for DELX and DELY are specified; however, TOPOG will shift the CG-MATHEW and ADPIC grids as much as DELX (or DELY, as appropriate) to ensure the overlaying of the proper points. The origins will be shifted as little as possible from the positions given in TOPOG.NML. The other constraint on the grids is that no part of the CG-MATHEW grid may lie outside of the terrain database grid, and that no part of the ADPIC grid may lie outside of the CG-MATHEW grid. If the ADPIC grid, as specified, lies within the terrain data base grid and the default CG-MATHEW grid does not, then TOPOG will attempt to shift the CG-MATHEW grid to create a legal system of grids (See Fig.II.C.1.).

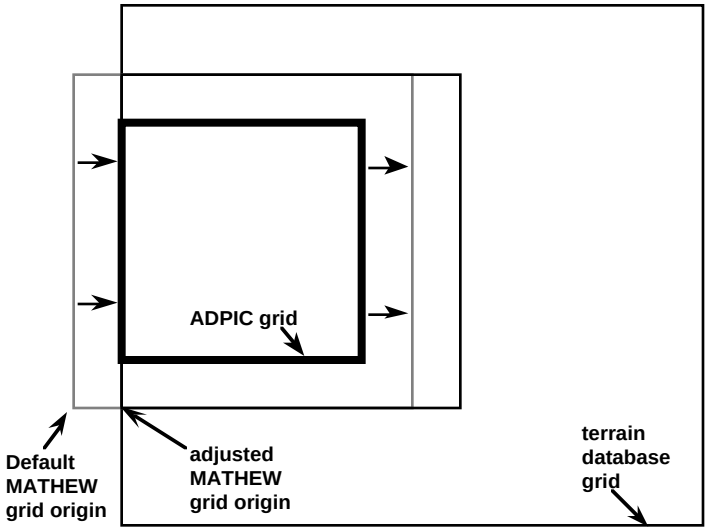


Fig. II.C.1. Example of the automatic shift of the CG-MATHEW grid by TOPOG to create a permissible system of grids.

The ADPIC grid can be specified as being as much as one DELX or DELY step out of the area covered by the regional elevation file grid. If so, the ADPIC grid will be shifted so that the appropriate boundary matches the elevation file grid boundary. The CG-MATHEW grid will also be shifted as described above to match the elevation file grid boundary along the corresponding border. If the ADPIC grid lies further from the border that one CG-MATHEW/ADPIC grid step, then an error is reported and TOPOG ends.

If there is no important terrain relief in an area or if no elevation data is available, TOPOG can generate a flat grid of arbitrary size (see FLAT_TOPO in Section III.B.1.). There are then no constraints on the CG-MATHEW and ADPIC grids with respect to the elevation data base grid. Regardless of the means of selecting a grid (or even if errors are discovered during validation), all grid parameters are displayed on the terminal and in the TOPOG.LOG file to allow the user to

determine the acceptability of the model grids produced by TOPOG, or to determine the cause of any errors.

The final CG-MATHEW grid, along with the vertical grid spacing (DELZ) selected by the user, defines a system of grid cells filling a grid volume. Terrain is represented by specifying which grid cells in a vertical column are open (wind is permitted to pass through at least one of the cell's faces) and which cells are closed (i.e., underground). The terrain is described by a two-dimensional array of integers with each element in the array specifying the integer number of grid cells that lie underground in the CG-MATHEW grid.

TOPOG generates the terrain arrays by averaging the data in the regional elevation data base over the CG-MATHEW grid. The standard elevation data file specifies elevations for a region covered by a grid of square cells oriented along the appropriate map coordinate system (typically UTM coordinates for regional problems). Elevation data base grids are typically dimensioned 400 by 400 with each grid cell commonly a 500 m square (in such cases the area described is 200 km by 200 km). The terrain file contains an array of values with each value being the average height, in meters above sea level, of the corresponding grid cell. Since the grid has been accepted by TOPOG before the averaging begins, each CG-MATHEW cell is an integer multiple of one quarter the elevation grid spacing and, as a result, the average height of a CG-MATHEW cell can be found by averaging the heights of the portions of the elevation data base grid cells it contains with the averaging weights reflecting the area of the portion of the elevation grid cell contained in the CG-MATHEW grid cell. On completing this step a height in meters above sea level is associated with each CG-MATHEW cell. In the past, the terrain averaging done in TOPOG was based on CG-MATHEW double cells, i.e., four CG-MATHEW grid cell in a 2 by 2 array were averaged as an entity. This was used to avoid single cell holes in the model terrain that permit ADPIC particles to diffuse into them but cannot contain any winds, since five of the six faces are closed, so such particles cannot be advected out of them. Current procedures in TOPOG, largely interactive, allow single cell holes to be removed in other ways (automatically if the code is being run non-interactively) so this option is rarely used in ARAC applications. However, TOPOG is occasionally ported to other systems that may not be able to support the necessary graphics tools, so the option for this double cell averaging is maintained (see DOUBLECELL in Section III.B.1.),

To complete the averaging, each cell height is rounded to the nearest integer multiple of DELZ. The lowest value for all the cells in the grid is then subtracted from each grid value, thereby re-defining the bottom of the CG-MATHEW grid volume. The integer height of each cell is then assigned to the appropriate CG-MATHEW grid cell. This array is written to the TOPOG*.GRID file for use by the remaining models. The ADPIC cell heights are extracted directly from this array when ADPIC reads the TOPOG*.GRID file, accounting for any offsets due to the typically larger CG-MATHEW grid dimensions. CG-MATHEW requires its terrain heights to be offset so that the bottom of the grid has a value of one. Hence, both MEDIC and CG-MATHEW increment each cell height by one when they read the TOPOG*.GRID file. A two-dimensional illustration of the terrain representation is shown in Fig. II.C.2.

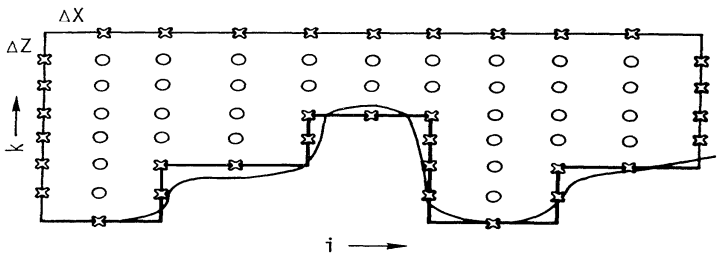


Fig. II.C.2. Illustration of the topography specification in two dimensions.

II.C.2. MEDIC

Since CG-MATHEW adjusts the extrapolated winds by the *minimum* necessary amount in producing a mass-consistent arrangement, a reasonable, first-guess field of wind velocities, $\underline{U}^0$ in Eq. (II.A.1), must be supplied by MEDIC, the wind extrapolation code. Attention to detail in the preparation of the extrapolated field is critical. Fairly minor variations in the parameters input to MEDIC can lead to grossly different wind fields, all of which are consistent with the measured data. For this reason, we rely heavily on the interpretive ability of an experienced meteorologist to determine the acceptability of a result. Using the tools of his trade (e.g., weather maps, local observations, upper air data, pilot reports), a meteorologist can factor in all of the known and relational information (in the form of model input) to produce a realistic and meteorologically consistent wind field. If necessary, the model is rerun with altered input until the desired result is achieved. This procedure is extremely important in that physical processes, many of which are not modeled in CG-MATHEW, can be implicitly represented in the structure of the extrapolated field.

The creation of an extrapolated gridded wind field from available wind observations is accomplished by one of two related algorithms. The first method uses surface observations and user-defined parameters to construct the required three-dimensional wind field with minimal upper air data required. This method is referred to as parameterized extrapolation; it is used when no upper air soundings are available or when the credibility of a sounding is in doubt. The other approach makes use of all surface and upper air observations in the wind extrapolation process. This approach is called profile extrapolation; it is the preferred method if at least one local upper air sounding, taken within the past six hours, is available. Parameterized extrapolation is the default in MEDIC (see (PROFILE_EXTRAP in Section II.B.4.).

Three separate layers of the atmosphere are recognized in the extrapolation procedures. The surface layer is the lowest layer in which surface effects predominate and no direction shear with height is allowed. The highest layer, the geostrophic layer, is the region of the atmosphere where surface effects have virtually dissipated and the winds reflect larger-scale synoptic flow. Wind speed and direction are assumed to be constant with height in this layer, i.e., there is no wind shear. The central layer, or boundary layer, is the region where conditions are intermediate between the surface and geostrophic layers; both speed and directional shear can occur.

Both extrapolation procedures produce two three-dimensional arrays of horizontal wind components, one array for the u-components and the other array for the v-components, containing the wind vectors at each grid point of a volume having the same dimensions as the CG-MATHEW grid. However, unlike the CG-MATHEW grid, the extrapolation grid volume is adjacent to but entirely above the model terrain surface (i.e., in the extrapolation grid, heights above grid bottom are always equal to heights above model terrain). On completion of the extrapolation, model terrain is pushed up through the grid bottom with a corresponding upward shift in the extrapolated winds. Thus, the atmospheric layers defined in the extrapolation process (Z_{sl} and Z_{bl}) are flat in the extrapolation grid volume and have the shape of the terrain in the CG-MATHEW grid volume (see Fig. II.C.4.). This produces the wind fields that are passed to CG-MATHEW. CG-MATHEW transforms the winds into face-centered components for ease of calculation (see Section II.A.3.).

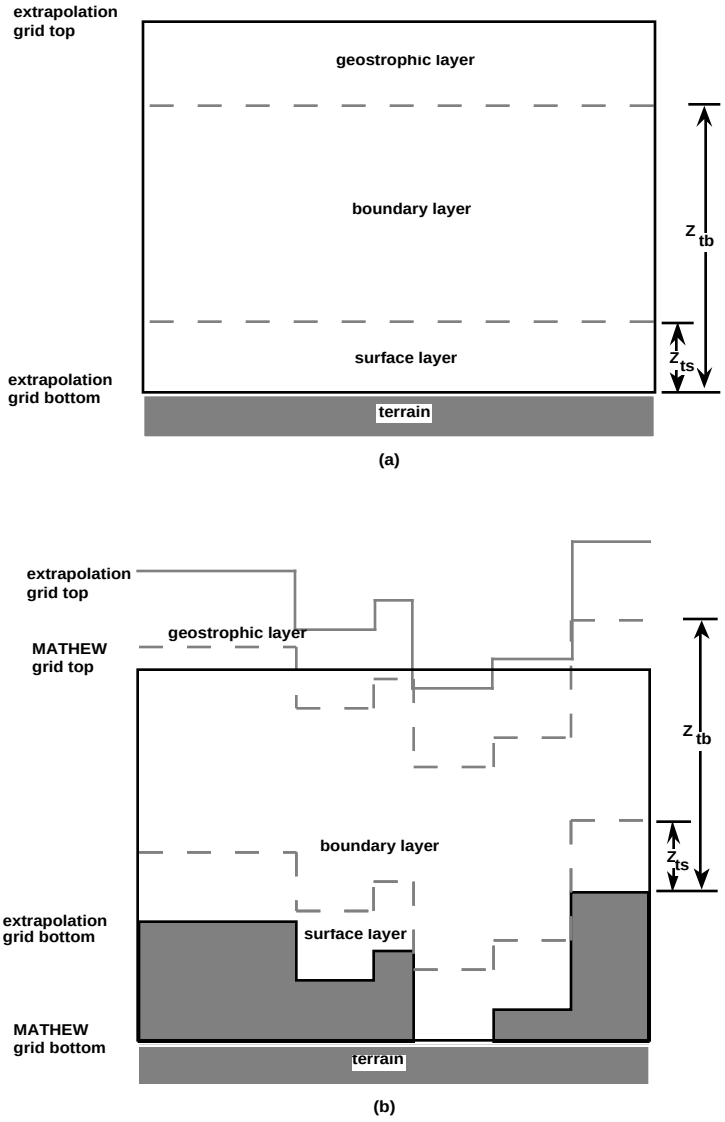


Fig. II.C.4. Schematic of a CG-MATHEW grid in two dimensions: (a) before, and (b) after the introduction of model terrain.

II.C.2a. Parameterized Extrapolation

Parameterized extrapolation involves two basic steps. First, wind speed and direction values must be assigned to two height levels in each column of grid points of the extrapolation grid volume. One level is near the surface and the other level is at or above the top of the grid. Second, winds along the columns of grid points are determined by interpolating between the wind values at the two levels, and by extrapolating to the surface below the lower level. The interpolations and extrapolations are controlled by a number of user-defined parmeters that aid in simulating the vertical structure of the atmosphere.

The near-surface level wind for each column of grid points is derived from surface wind observations. First, all surface speed measurements are normalized to a user-specified height above terrain.called the reference height (see REF_HGT in Section III.B.5.). In order to minimize the alteration caused by normalization, REF_HGT is usually chosen to be the height above terrain of most of the observing instruments. The surface layer power law formula used in the normalization is

$$S_{\text{norm}} = S_{\text{obs}} \left(Z_{\text{ref}} / Z_{\text{obs}} \right)^{\gamma_{\text{sl}}} \quad , \tag{II.C.1}$$

where S_{norm} is the wind speed at height REF_HGT, S_{obs} is the observed wind speed, Z_{ref} is the reference height, Z_{obs} is the observation height, and γ_{sl} is the power law exponent for the surface layer (see PWR_LAW_EXPNT_SL in Section III.B.5. and the description of the OBSERV.MET file in Section III.B.7). Wind directions are not altered when normalization occurs since directions rarely show significant changes for typical adjustment heights. The reference level speed and direction for a column of grid points are then derived from the normalized measurements by an extrapolation based on the inverse square of the distance to nearby observation points according to the formula:

$$S_{\text{ref}} = \frac{\sum_{n=1}^{N} S_n / R_n^2}{\sum_{n=1}^{N} 1 / R_n^2} \tag{II.C.2}$$

where S_{ref} is the reference level speed for the grid point column, N is the number of nearest observations being used in the extrapolation, S_n is the normalized speed measurement for a given station, and R_n is the distance from the horizontal location of the column to the station. The four nearest stations are normally used, although this can be changed by recompiling. The reference level direction for the column is calculated by separate inverse-square extrapolations of the sines and the cosines of the observed values based on formulas analogous to Eq. (II.C.2.). The sines and cosines are extrapolated separately to avoid the discontinuity caused by interpolating through 0˚ (360˚). and are then used to generate a consistent angle using an arctangent function. The subset of the observation stations used to determine the direction may not be the same as the subset for speed since a station may report only speed or direction. Therefor the selection of the nearest observation stations is performed separately for the speed and direction extrapolations. At least one surface speed and one surface direction observation, not necessarily at the same point, must be input to allow the extrapolation to proceed.

The upper level wind in a grid point column is generated from upper air wind observations. The upper level is usually taken to be the top of the extrapolation grid volume. However, if the top of the boundary layer is above the top of the extrapolation grid, the wind is instead determined at the top of the boundary layer. In order to extract values at arbitrary heights from the discrete upper air profile measurements provided by a sounding, it is assumed that winds vary linearly between successive measurement heights. Directions vary through an arc describing the smallest angle between two measurements. Above the highest measurement level, winds are assumed to remain constant with height Winds below the lowest measurement level are taken to be constant down to the top of the surface layer and to obey a power law for speed within the surface layer. The inverse square extrapolation procedure [see Eq. (II.C.2)] is then used to assign a speed and direction at the upper level of each grid point column. Profiles with speed measurements only or direction measurements only will be accepted as input. At least one upper air speed measurement and one upper level direction measurement, not necessarily at the same location, must be input to allow the extrapolation to proceed.

Having extrapolated winds at two levels, values can now be assigned to all the grid points in a column using an interpolation method that reflects the basic three layer structure of the atmosphere. For grid points in the surface layer above and below the reference height, the wind speed is determined using a power law formula similar to that used for the reference level normalization (Eq.II.C.1) :

$$S(Z) = S_{\mathrm{ref}} \left(Z / Z_{\mathrm{ref}} \right)^{\gamma_{\mathrm{sl}}} \qquad \text{(II.C.3)}$$

where S is the speed at a surface layer grid point, Z is the height of the surface layer grid point, and S_{ref} and Z_{ref} are the reference level speed and height for the column of grid points. Since there is no wind shear in the surface layer, the reference level direction is assigned to all surface layer grid points of the column. The power law formula is also used to generate a surface slip velocity at the surface roughness height, Z_{srh} (see SFC_ROUGH_HGT in Section III.B.5.). In the geostrophic layer, constant winds are assumed and the upper level speed and direction are assigned to all grid points in the column above the boundary layer.

Winds in the boundary layer are computed so as to provide continuity between the surface layer winds and the geostrophic layer winds. The simplest way of satisfying this condition is to linearly interpolate between the winds at the top of the surface layer and the top of the boundary layer. This capability is provided and is adequate for many situations. However, atmospheric conditions can arise which require greater generality in the interpolation. The speed shear in the boundary layer is often variable, with the greatest change occurring near the top, as frequently occurs when the top of the boundary layer coincides with a temperature inversion. This behavior may be modeled by a power law interpolation formula with a user-defined exponent. Another important consideration in the boundary layer is the variation of direction with height. The difference between any two angles can be determined along two directions of rotation, clockwise or counter clockwise. As a result, there are two possible interpolations between the direction at the top of the surface layer and the direction

at the top of the boundary layer. In middle latitudes of the northern hemisphere, winds usually change in a clockwise direction with increasing height, i.e., they "veer" with height. Therefore, it is not always advisable to determine angular difference through the smallest angle. There are also cases where the wind directions turn counter clockwise with height, i.e., they "back." This can occur, for example, when a front moves through an area. It is important for an assessor to be able to control the sense of the directional shear in the boundary layer, a capability which is provided in MEDIC.

Interpolation in the boundary layer is accomplished using the following formulas:

$$S(Z) = \frac{\Delta S_{bl}\left(Z - Z_{ts}\right)^{\gamma_{bl}}}{\left(\Delta Z_{bl}\right)^{\gamma_{bl}}} + \text{Sts} \ , \qquad \text{(II.C.4a)}$$

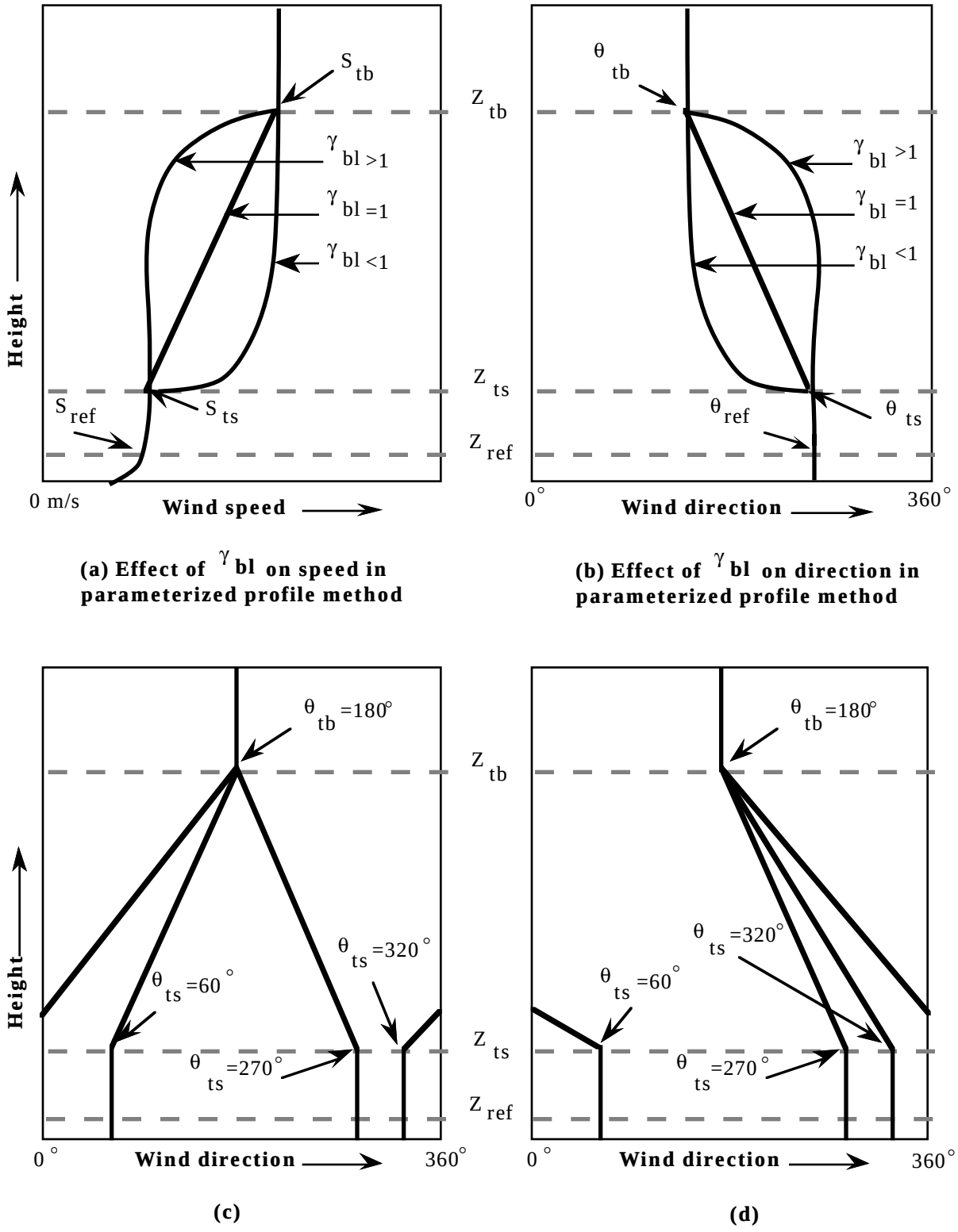
$$\theta(Z) = \frac{\Delta'\theta_{bl}\left(Z - Z_{ts}\right)^{\gamma_{bl}}}{\left(\Delta Z_{bl}\right)^{\gamma_{bl}}} + \theta\text{ts} \qquad \text{(II.C.4b)}$$

where γ_{bl} is the boundary layer extrapolation exponent (see EXTRAP_EXPNT_BL in Section III.B.5.). ΔS_{bl} is the speed shear over the depth of the boundary layer, i.e., $\Delta S_{bl} = S_{tb} - S_{ts}$ where S_{tb} and S_{ts} are the speeds at the top of the boundary layer and the top of the surface layer, respectively. ΔZ_{bl} is the depth of the boundary layer, i.e., $\Delta Z_{bl} = Z_{tb} - Z_{ts}$ where Z_{tb} and Z_{ts} are the top of the boundary layer and the top of the surface layer heights, respectively (see BL_HGT and SL_HGT in Section III.B.5.). $\Delta'\theta_{bl}$ is the direction difference over the depth of the boundary layer. This difference depends on θ_{tb} and θ_{ts}, the directions at the top of the boundary layer and the surface layer, respectively, and θ_r (see MAX_VEER_PARAM_VERT in Section III.B.5.) according to the following recipe:

$$\Delta\theta = \theta_{tb} - \theta_{ts} \ . \qquad \text{(II.C.5)}$$

IF $\theta_r - 360^\circ \le \Delta\theta \le \theta_r$, then $\Delta'\theta_{bl} = \Delta\theta$;
if $\Delta\theta < \theta_r - 360^\circ$, then $\Delta'\theta_{bl} = \Delta\theta + 360^\circ$,
if $\Delta\theta > \theta_r$, then $\Delta'\theta_{bl} = \Delta\theta - 360^\circ$.

The assignment of a value to θ_r causes wind directions to veer with height if the directional shear, determined in the clockwise sense, is less than θ_r ; it causes wind directions to back with height if the clockwise shear is greater than θ_r. Equations (II.C.4a) and (II.C.4b) are evaluated at the heights of all the grid points within the boundary layer. When the entire process described above is completed for every column of grid points in the grid, the extrapolation volume is ready to be examined by the assessor. The assessor can adjust the parameters in the extrapolation $\left(Z_{srh}, Z_{ref}, Z_{ts}, Z_{tb}, \gamma_{sl}, \gamma_{bl}, \text{ and } \theta_r\right)$ if the extrapolated field does not provide a reasonable picture of the atmosphere. The effects of two of the parameters, γ_{bl} and θ_r are shown in Figs.II.C.5a-d.



(a) Effect of γ_{bl} on speed in parameterized profile method

(b) Effect of γ_{bl} on direction in parameterized profile method

(c)

(d)

Fig. II.C.5. Effect of (c) θ_r = 240˚ and (d) θ_r = 100˚ on various values of θ_{ts} in parameterized profile method.

As mentioned previously, the parameterized extrapolation method is normally used only in the absence of acceptable upper air data. When upper air data is unavailable for a site, an estimate of the upper level winds must be supplied. A reasonable evaluation of the wind speed and direction might be extracted from synoptic charts for an appropriate height. If no other basis can be found for a selection, climatological data might be helpful in providing an estimate. If a recent, nearby upper air sounding is available, then the profile extrapolation method should be used because this procedure takes greater advantage of the available information.

II.C.2b. Profile Extrapolation

The profile extrapolation method provides a mechanism for the controlled melding of surface and upper air wind observations controlled by the same variables used in the parameterized method, although some of the variables are used in slightly different ways. It is assumed that surface observations provide the most reliable picture of the actual winds in the lower levels of the atmosphere and should be weighted more heavily than upper air observations in any interpolation scheme since the density of surface stations is almost always much higher than the density of upper air stations. For a typical vertical grid dimension, an upper air sounding provides the best wind information for extrapolation near the top of the grid, while at intermediate grid point levels, the relative importance of the surface and upper air observations will vary with the meteorological conditions. It is also assumed in this method that the "shape" of a measured profile, i.e., its general features, can be applied to different locations in a region, even if the actual speed and direction values at a given level do not match those for the other positions.

Three main steps are required in the profile extrapolation method. First, a wind speed and direction derived from the surface observations must be assigned to each column of grid points in the CG-MATHEW grid at a level near the bottom of the extrapolation grid. Next, each grid point in a column is assigned speed and direction values which are derived entirely from the upper air soundings, i.e., a profile is synthesized for the grid point column from the upper air data alone. Finally, the surface wind is combined with the synthesized profile to give an extrapolated wind at each grid point in the column. The surface and upper air components are weighted differently at different heights, with the weighting being controlled by various user-defined parameters.

The near surface wind at the reference height Z_{ref} is generated from surface observations using the same procedure as in parameterized extrapolation. For grid points below Z_{ref} the wind speed is found according to Eq. (II.C.3) and the slip velocity is found by applying Eq. (II.C.3) at the surface roughness height.

A synthesized profile is created along the column of grid points by inverse square extrapolation of the nearest upper air observations at the same height (Eq.II.C.2). It consists of wind speeds and directions for each grid point in the column $S_{\text{syn}}(Z)$ and $\theta_{\text{syn}}(Z)$, as well as for the reference level Z_{ref}, the top of the surface layer (Z_{ts}), and the top of the boundary layer (Z_{tb}).

In general, the near-surface wind derived from the surface data will not match the winds derived from upper air data. The two are meshed by implementing a height dependent correction term which is added to the synthesized profile value at each grid point in the column:

$$C_s(Z) = \frac{-\Delta S_{ref}}{Z_{tb} - Z_{ref}}(Z - Z_{ref}) + \Delta S_{ref} \text{ for } Z_{ref} \le Z < Z_{tb} , \quad \text{(II.C.6a)}$$

$$C_s(Z) = 0 \qquad \text{for } Z \ge Z_{tb} , \quad \text{(II.C.6b)}$$

where Z_{tb} and Z_{ref} are the heights of the boundary layer and the reference level, and Z is the height above terrain of the point under consideration. ΔS_{ref} is the discrepancy between the reference level wind speed and the speed from the synthesized profile at the same height, i.e., $\Delta S_{ref} = S_{ref} - S_{syn}$ at Z_{ref}. The extrapolated wind speed for a grid point in the column are given by:

$$S(Z_k) = S_{syn}(Z_k) + C_s(Z_k) \qquad \text{for} \qquad Z_k \ge Z_{ref} \quad \text{(II.C.7)}$$

where Z_k is the height of the grid point of interest. From examination of Eqs. (II.C.6) and (II.C.7), it can be seen that when $Z = Z_{ref}$, then $C_s(Z_{ref}) = \Delta S_{ref}$ and $S(Z_{ref}) = S_{ref}$. When $Z \ge Z_{tb}$, then $C_s(Z) = 0$ and $S(Z) = S_{syn}(Z)$. The correction term varies linearly between Z_{ref} and (Z_{tb}), and goes to zero at the top of the boundary layer above which the extrapolated profile merges with the synthesized profile. A sample extrapolation for a column of points is shown in Fig. II.C.6.

The directional extrapolation procedure is similar to that for the speed. Upper air observations have no effect on the surface layer. Therefore, since there is no directional shear in the surface layer, the reference level direction, θ_{ref}, is used for all grid points below Z_{ts} (it is always assumed that the reference height is within the surface layer). The effect of upper air soundings begins at the top of the surface layer and increases with height in the boundary layer until they completely dominate in the geostrophic regime. The directional shear in the intermediate boundary layer is controlled by the same parameters as in parameterized extrapolation (γ_{bl} and θ_r). The correction term is related to the discrepancy between the synthesized direction, based on upper air data, at the top of the surface layer and the reference level direction used throughout the surface layer. The correction term

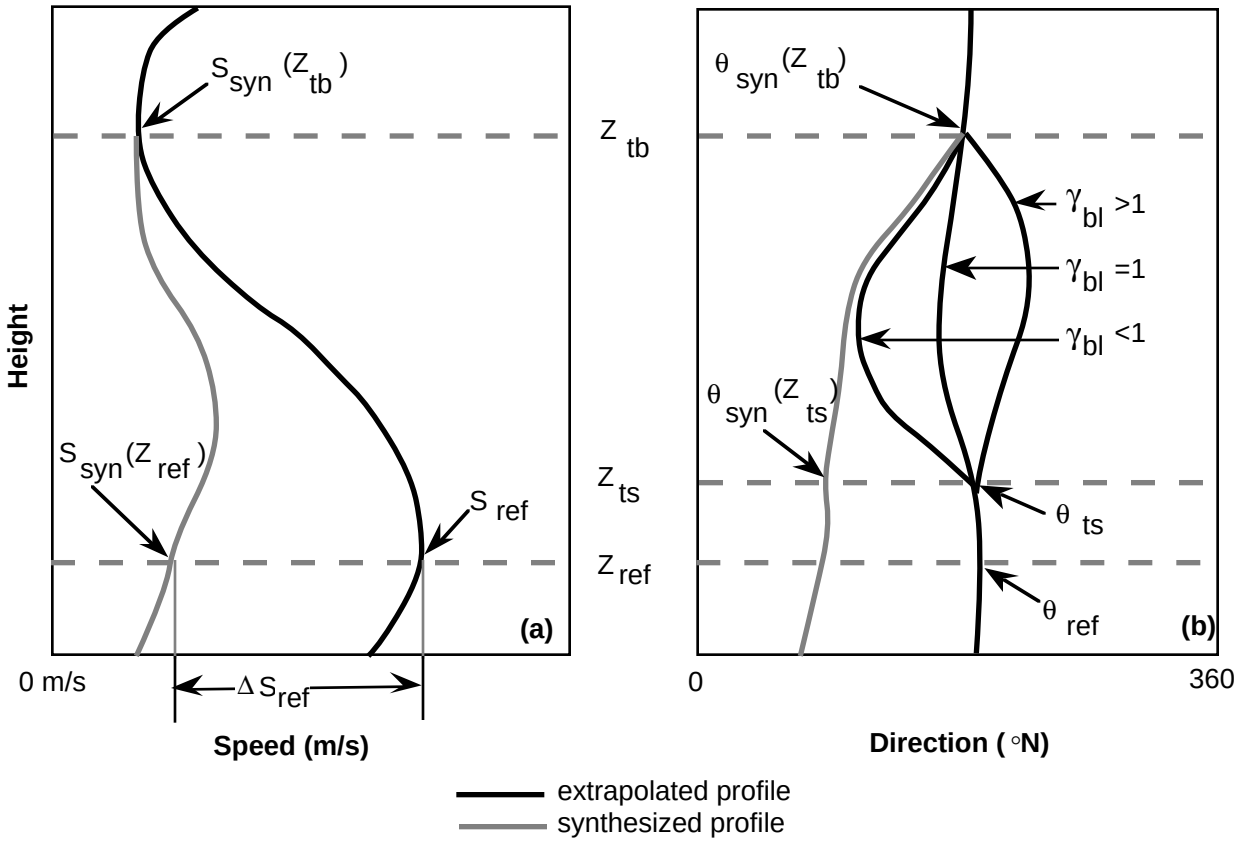


Fig. II.C.6. Examples of (a) speed and (b) direction extrapolation (as a function of γ_{bl}) using the profile method.

for direction is given by:

$$C_\theta(Z) = \frac{\Delta'\theta}{(\Delta Z_{bl})^{\gamma_{bl}}}(Z - Z_{ts})^{\gamma_{bl}} - \Delta'\theta_{ts} \text{ for } Z_{ts} \le Z < Z_{tb} \quad , \tag{II.C.8a}$$

$$C_\theta(Z) = 0 \qquad \text{for} \qquad Z \ge Z_{tb} \quad , \tag{II.C.8b}$$

where ΔZ_{bl}, Z_{tb}, Z_{ts}, and γ_{bl} have the same definitions as in the parameterized method, Z is the height of the point under consideration, and $\Delta'\theta_{ts} = \theta_{syn}(Z_{ts}) - \theta_{ts}$ is the direction discrepancy at the top of the surface layer between the surface observations and the profile synthesized from the upper air data. The sense of how this discrepancy is resolved (via backing or veering) is controlled by θ_r in a manner analogous to Eq. (II.C.5), except that $\Delta\theta = \theta_{syn}(Z_{ts}) - \theta_{ts}$. It is important to note that θ_r in the profile method is used for comparing angles at the top of the surface layer, while θ_r in the parameterized method is used to compare angles at two different vertical levels (the extremes of the boundary layer). The extrapolated wind for a grid point in the column is given by

$$\theta(Z_k) = \theta_{syn}(Z_k) + C_\theta(Z_k) \qquad \text{for} \qquad Z_k \ge Z_{ts} \quad , \tag{II.C.9}$$

where Z_k is the height of the grid point of interest. From examination of Eqs. (II.C.8) and (II.C.9), it can be seen that $\theta(Z_{ts}) = \theta_{ts}$ and $\theta(Z_{tb}) = \theta_{syn}(Z_{tb})$. The correction term varies between Z_{ts} and Z_{tb} as a function of γ_{bl}. If $\gamma_{bl} = 1$, then the influence of the surface observations goes linearly to

zero at the top of the boundary layer. If $\gamma_{bl} \gg 1$, the influence of the surface observations is maintained throughout most of the boundary layer before vanishing at the top. If $\gamma_{bl} \ll 1$, then the synthesized profile dominates the extrapolated winds through most of the boundary layer. The effects of selected choices for γ_{bl} are shown in Fig. II.C.6.b. When the process described above is completed for each column of points in the extrapolation grid volume, these winds can be examined by the assessor. As in the parameterized case, all the user-defined parameters in the process are available for adjustment if the extrapolated wind field fails to give an acceptable description of the atmosphere.

II.C.2.c. Terrain Incorporation

In both methods the final products are grids of u- and v-component winds The extrapolated wind arrays are transformed to the CG-MATHEW grid volume by the incorporation of terrain which is controlled by the TERRAIN_ADJUSTMENT variable. If set to 'N' (the default for hemispheric MEDIC), the winds are zeroed up to the terrain level, defined as the highest of the (usually four) cell-centered terrain levels adjacent to a grid point column.as generated by TOPOG. If TERRAIN_ADJUSTMENT is set to 'Y' (the default for regional MEDIC), each column of grid points is shifted upwards by the number of grid points along the column that are at, or below, the block terrain surface in the CG-MATHEW grid volume. Slip velocities are then assigned along vertical faces of the terrain surface using the u- and v-components of the slip velocity at the bottom of the extrapolation grid volume. Components that are tangential to a vertical terrain surface are assigned to the related grid point unless, at the same point, the component is normal to another vertical terrain surface. Once terrain incorporation has been performed, the winds are ready to be passed to CG-MATHEW.

II.C.2.d. Gridded wind fields

Upper air data available in the form of AFGWC gridded metdata files may be used by MEDIC (see MEDIC_PARAMS namelist parameter NHEDATE, NHETIME, NHEFCST, SHEDATE, SHETIME, and SHEFCST). The AFGWC files contain analysis or forecast data and cover either the northern or southern hemisphere to within 20 degrees of the equator. The upper air metdata data is in the form of u- and v- component gridded winds on constant pressure surfaces of dimension. The AFGWC grids are dimensioned 47 by 51 by 15. MEDIC outputs tables of the valid wind data for each AFGWC file input.

Gridded metdata from the Fleet Numerical Meteorological and Oceanographic Center's (FNMOC) Navy Operational Global Atmospheric Prediction System (NOGAPS) may also be used by MEDIC (see MEDIC_PARAMS namelist parameters NODDSDATE, NODDSTIME, and NODDSFCST). Currently these data arrive routinely via the internet, and they can also be retrieved using the Navy Operational Data Distribution System (NODDS). In either case, special interim procedures must be used to prepare the data for use in MEDIC. An automated, permanent,

operational method to acquire and prepare NOGAPS data is expected to be completed by December 1995.

II.C.2e. Hemispheric MEDIC (HMEDIC)

The hemispheric version of MEDIC is built on a 47 by 51 by 15 grid to match the AFGWC grids using a polar stereographic projection. Winds input from AFGWC upper air data (see Part II.C.2.d) are linearly interpolated to the hemispheric computational grid and the appropriate map projection correction is applied. At least two pressure levels of u- and v- data must be present for continued processing of the AFGWC winds from a given file. Otherwise a warning is printed and the code skips to the next data set. Terrain may be incorporated by zeroing winds up to, but not including, the terrain surface (see Part II.C.2.c).

While the hemispheric version of MEDIC will only utilize the gridded metdata from AFGWC, the regional version of MATHEW/ADPIC can utilize metdata from a variety of combinations of different sources, including AFGWC gridded metdata, as well as FNMOC gridded metdata. Therefore, with the use of a generalized map projection capability, it is now possible to use the "regional" MATHEW/ADPIC models to model any area from a few kilometers up to global scale. It is expected that the special hemispheric version of the MATHEW/ADPIC models will cease to be utilized by ARAC Operations in the near future, since with the appropriate map projection, and the right input parameter defaults for the various scales, the "regional" version has as much capability and much more flexibility than the hemispheric version.

II.C.3. PLCNT

PLCNT is a code for processing the concentrations produced by ADPIC. The ADPIC output is passed to PLCNT in CONC*.BIF (see Section III.C.10), a file that contains one concentration grid set for each sample bin in ADPIC. A concentration grid set includes a main grid, matching the ADPIC advection grid, along with up to five nested grids each with a dimension equal to one half the main grid and a grid step also equal to half the main grid step. A CONC*.BIF file is created at each ADPIC output time, where the output interval is specified by SAMPLING_INTERVAL.

Concentrations can be sampled in three ways: instantaneous, time integrated, and total deposition. (Future plans include addition of a fourth way: integrated deposition.) Contours of instantaneous air can be considered "snapshots" of concentration (or dose) at a given point in time and at a specific level above the surface. Contours of integrated air depict the total amount of pollutant passing a given point during a specified time interval. If a dose conversion factor is specified, then the contours represent the total dose to an unshielded human from exposure to a plume or puff. Contours of deposition show distributions of material deposited since the beginning of the release to the time of output. In an ARAC context, results are generally expressed as micrograms per square meter $\left(\mu g/m^2\right)$, curies per square meter $\left(Ci/m^2\right)$, or a dose to the skin or through inhalation or ingestion.

To produce a desired plot, the user must appreciate the interplay between certain parameters in ADPIC and PLCNT. The critical parameters in ADPIC are SOURCES_TO_SAMPLING_BIN (the sources to be associated with a sampling bin, by number), SAMPLING_TYPE (the type of concentration sampling), and SAMPLING_HGT (the sampling height for integrated and instantaneous air sampling). These arrays are all dimensioned maxsbins (currently 30) with corresponding elements of each array referring to one of the concentration grid sets (i.e. a main grid and up to four nested grids) that define a sampling bin in ADPIC. Thus, the first element of SOURCES_TO_SAMPLING_BIN indicates which of the MAXSOURC sources is to be considered in the first ADPIC bin; the first element of SAMPLING_TYPE indicates whether the first ADPIC sampling bin is to be integrated, instantaneous, or deposition; the first element of SAMPLING_HGT indicates the height above terrain at which the first ADPIC sampling bin is to be placed. The second elements of each array describe the second sampling bin, and so on.

PLCNT can be thought of as having a dose or units conversion function and a contour plotting function. The three parameters that control the dose conversion are SAMPLING_TO_DOSE_BIN, DCON, and DOSE_TO_FINAL_BIN, which are contained in the CONC_ANALYSIS.NML file. The first two parameters allow the user to combine bins to produce combined nuclide dose plots (e.g., combined dose from multiple nuclides or multiple hybrid sources), and to re-use bins to produce different contour plots from the same ADPIC sampling bin (e.g., thyroid dose and effective WB dose derived from integrated air concentration). SAMPLING_TO_DOSE_BIN specifies how the sampling bins are to be assigned to the dose bins; DCON specifies the dose conversion factors and/or scaling

factors to be applied to the dose bins (also see description for DCON in Section III.A.8); DOSE_TO_FINAL_BIN specifies how the dose bins are to be assigned to the final bins. Examples and their translations are shown in Figure II.C.7.

A	SPECIES	='I-131'	'Kr-88'	'Pu-239'		
	SOURCES_TO_SAMPLING_BIN(1,1)	=1				
	SOURCES_TO_SAMPLING_BIN(1,2)	=2				
	SOURCES_TO_SAMPLING_BIN(1,3)	=3				
	SAMPLING_TYPE	=1	1	1		
	SAMPLING_HGT	=1.5	1.5	1.5		
B	SAMPLING_TO_DOSE_BIN	=1	1	2	3	3
	DCON	=3.60E2	7.63	3.65E-1	1.13E5	3.96E5
C	DOSE_TO_FINAL_BIN(1)	=1				
	DOSE_TO_FINAL_BIN(2)	=2	3	4		
	DOSE_TO_FINAL_BIN(3)	=5				

Translation of A:

Sampling bin #1 will contain integrated (at 1.5m height) air concentration of I-131 source.
Sampling bin #2 will contain integrated (at 1.5m height) air concentration of Kr-88 source.
Sampling bin #3 will contain integrated (at 1.5m height) air concentration of Pu-239 source.

Translation of B:

Dose bin #1 will contain sampling bin #1 converted to I-131 Thyroid dose via inhalation.
Dose bin #2 will contain sampling bin #1 converted to I-131 Effective WB dose via inhalation.
Dose bin #3 will contain sampling bin #2 converted to Kr-88 Effective WB dose via immersion.
Dose bin #4 will contain sampling bin #3 converted to Pu-239 Effective WB dose via inhalation.
Dose bin #5 will contain sampling bin #3 converted to Pu-239 lung dose via inhalation.

Translation of C:

Final bin #1 will contain dose bin #1 (data for thyroid dose plot from I-131).
Final bin #2 will contain dose bins #2, #3, #4 (data for Effective WB dose plot for three nuclides).
Final bin #3 will contain dose bin #5 (data for lung dose plot from Pu-239).

Fig. II.C.7. Example of interplay between ADPIC and PLCNT parameters to produce desired types of plots.

The times of contour plot generation are determined by defining one or more plot series for each final bin of interest. A plot series has a beginning and an ending point in time along with a time step that specifies how ofter a plot is to be generated. The first plot is produced at the beginning of the plot series (which must match the time of a CONC*.BIF file) and subsequent plots in the series are produced at equal intervals after that until time steps beyond the end of the plot series. In the case of interval integrated air plots, an integration step may be specified that may be less than the plot step. In such cases. the integrated plot will include concentrations beginning one integration step before the end of the current plot step, and thus, the integration period will end at the time of the next plot. Plot series are started with the variables PLOT_START_DATE/TIME and explicitly ended with

PLOT_END_DATE/TIME. If a PLOT_END_DATE/TIME is not specified then the current plot series is ended when another plot series for the same final bin begins or by the end of the available concentration data in the series of CONC files. The plot step is specified by PLOT_STEP and the integration step by INTEGRATION_STEP. All these variables are described in more detail in Section III.B.9.

PLCNT allows the user to select up to maxcontlevel (currently 5) contour levels using the CONTOUR_LEVELS variable (see Section III.B.9). PLCNT will also generate contour levels in the absence of user-specified values. The code will select the largest value of the form 1.0En or 3.0En that is lower than the highest grid point concentration value for a given final bin (if HALF_POWER_OF_10 is set to 'N' only values of the form 1.0En will be selected). This becomes the value of the innermost contour. For example, if the maximum grid point concentration is 6.9E8, then the innermost contour will be 3.0E8. If the maximum grid point concentration is 2.4E8, then the innermost contour will be 1.0E8. Up to four more contours, each decreasing by an order of magnitude, are then computed to fill out the maximum desired number of contours (specified by MAX_CONTOURS). The above example strictly applies only if no restrictions are made on the minimum area to be covered by the maximum contour. (See the description of MIN_CONTOUR_AREA in Section III.B.9 for more information). If the user selected contour values but the concentrations are too small to produce a contour with area greater than MIN_CONTOUR_AREA, then PLCNT will optionally compute contours based on the maximum concentration (see RECOMPUTE_USER_CONTOURS and NUM_RECOMPUTED_CONTOURS in Section III.B.9).

ADPIC produces concentrations at grid points. PLCNT smooths the doses maintained in the final bins along the edges of the nested grids, thereby incorporating some influence of the values in the coarser grid surrounding the nest. This avoids almost all discontinuities in the contour lines when they cross nest boundaries. This smoothing is not required for the X-Windows versions concentration plotting codes and therefore the site workstation products are written from the final bins before these bins are smoothed.

Since ARAC provides a service to more than one customer, and each customer has slightly different needs, limited flexibility of plot layout has been allowed for, primarily focused on the lower half of the plot legend. In addition, the exact wording contained in almost all of the legend is also under user control. The layout and wording of the contour plot is controlled by a multitude of parameters. Figs. II.C.8, II.C.9, and II.C.10 show examples of three of the more common layouts and the values of the PLCNT parameters which control the resulting appearance.

Fig. II.C.8. Example of plot layout for DOD plots and PLCNT parameters which control the layout and wording.

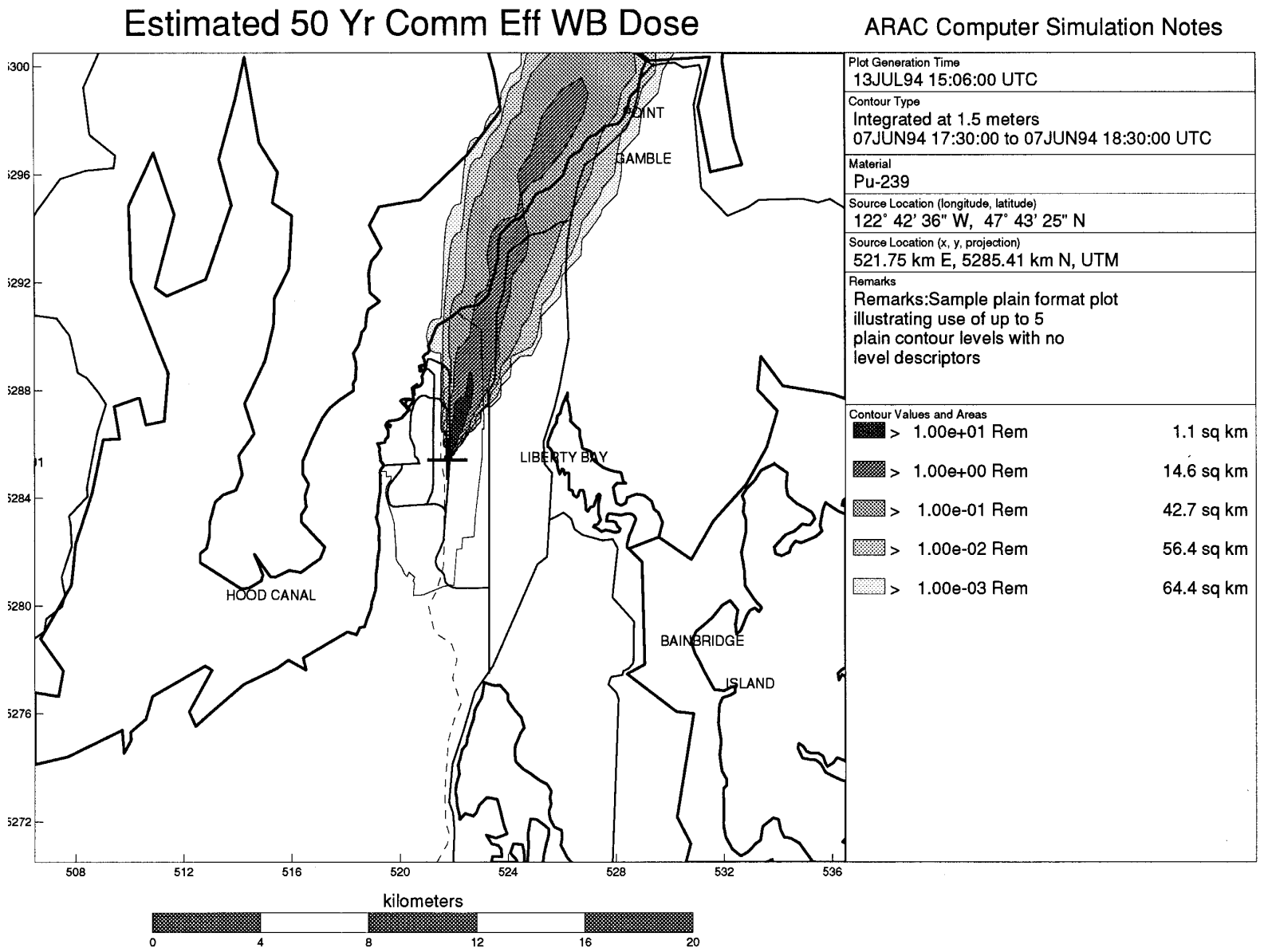


Fig. II.C.9. Example of plot layout for plain plots and PLCNT parameters which control the layout and wording.

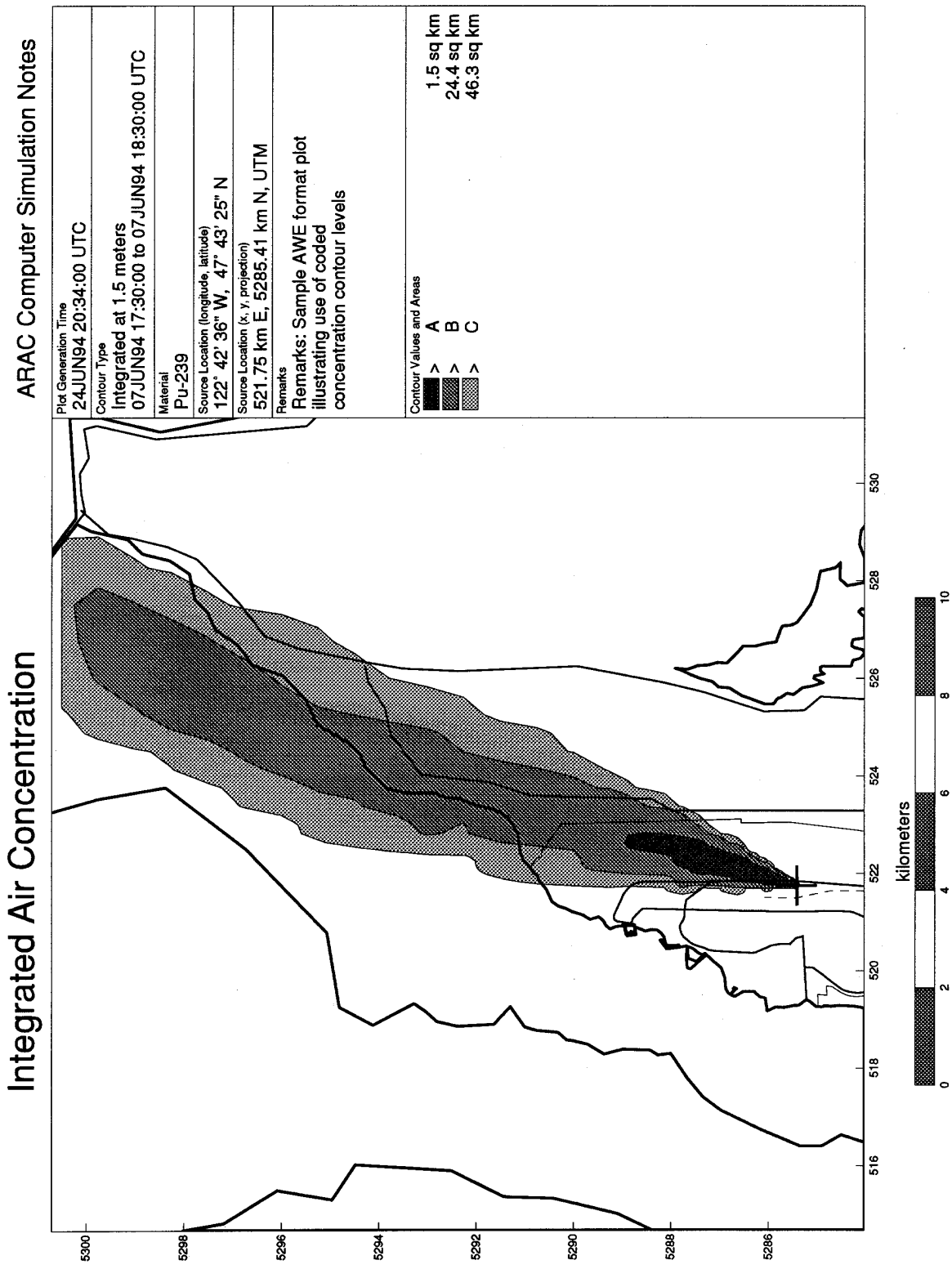


Fig. II.C.10. Example of plot layout for AWE plots and PLCNT parameters which control the layout and wording.

The ability to specify the area to blow up (or "window") for plotting is provided by the XMIN_WINDOW, XMAX_WINDOW, YMIN_WINDOW, and YMAX_WINDOW parameters. These four parameters specify the coordinates of the area to be blown up. Be aware that the areas specified in the legend of a blown-up plot refer to the areas covered by the contours in the original-size plot.

The ability to automatically generate a mosaic of subplots covering a specified area at a specified scale is provided by the following parameters: MAP_SCALE_FLAG, MAPSCALE, SOURCE_OCTANT, BLOWUP_SOURCE, PLOT_SIZE_CM. Up to 225 subplots (15 by 15) can be generated for a specified scale and output device. The placement of the ADPIC source location in the subplot containing it is controlled by SOURCE_OCTANT. The area to be split into subplots can be narrowed down by use of the blowup parameters mentioned in the previous paragraph.

The centerline capability of PLCNT starts at a user-specified ADPIC source location and searches along arcs of increasing radius for the maximum along each radial arc. The resultant series of maxima, along with their locations are available in tabular format in the CENLINE*.DAT file, and the locations are plotted on each Site Workstation plot. All centerline parameters are contained in the CONC_ANALYSIS_CONTROL namelist: CENTERLINE_SOURCE is the only required parameter; optional parameters include RADIAL_STEP_FACTOR, FULL_SEARCH_FACTOR, ANGLE_WIDTH_FACTOR, SECTOR_ANGLE_MIN, and SECTOR_ANGLE_MAX.

II.C.4. TIMEHIS

The Time History (TIMEHIS) code is an offspring of the PLCNT code. Concentration values are binlinearly interpolated for specific point locations. The times of the interpolations are controlled in the same way as in PLCNT and are tabulated in the TIMEHIS.LOG file as well as be plotted on a semi-log plot (log concentration vs. time). The number of data points per plot is controlled by the TIMESTEPS_PER_PLOT parameter. Interpolated points with values of zero are considered to be 1.0e-30 by the plotting routine so those values are plotted off scale to the bottom of the logarithmic vertical scale of the time history plot.

If actual measurement values are available, these values can also be input via the MEASUREMENTS section of the SAMPLER.DATA file, and appear on the plots along with the model-calculated values for the specified locations and times. Some additional statistics, based on the most straightforward ones used in the ATMES study, are also performed to compare the computed and measured concentrations. In addition to the TIMEHIS.LOG, which contains listings of the same time history data that is plotted, TIMEHIS produces two other ASCII files: SAMPOUT.DAT and STATS.DAT. SAMPOUT contains the interpolated value at each point along with the minimum and maximum point concentrations found within angles of various sizes (hit angles) of the main interpolation point. That is, concentrations are interpolated at a grid of points (currently 11 by 11)

ranging from -*dtheta* to +*dtheta* about angle *theta* and -*dtheta***radius* to +*dtheta***radius* centered on (*radius*,*theta*) where *radius* and *theta* are the polar coordinates of the main interpolation point with respect to a user selected source location and *dtheta* is specified hit angle (see HIT_ANGLES and STATISTICS_SOURCE in Section III.B.9). These minima and maxima are compared against measurements and if the measurement lies between the two values it is considered an exact match at that angle. The smallest hit angle is considered to be zero so the main point interpolated value is found listed in the first entry in the hit angle table (the minimum and maximum will always be the same for this entry). The STATS.DAT file contains the hit angle statistics along with some other simple statistics based on those used in ATMES.

II.D. MATHEW/ADPIC Validation and Evaluation

Because of the expense in computer time, the complexity of the three-dimensional transport and diffusion models, and the ensuing large number of variable parameters involved, no comprehensive error analysis of MATHEW/ADPIC has yet been undertaken. However, a number of tracer validation studies have been conducted and others are still in progress.

The ADPIC code using the RDM was validated against analytic solutions to the advection-diffusion equation (Ermak *et al.*, 1995) for both homogeneous and inhomogeneous turbulence. Numerical error was found to be less than 5% in these comparisons to analytic solutions. Over the last twenty years the MATHEW/ADPIC codes were also evaluated against numerous field tracer release experiments conducted under a variety of meteorological and topographical conditions (Foster and Dickerson, 1990). Recent work indicates that these evaluations are still representative of errors that can be expected when the RDM option in ADPIC is used. Neutrally buoyant tracers released both as surface and elevated point sources have been studied. A summary of the results is shown in Fig. II.D.1.

The results show that the MATHEW/ADPIC models generally estimate the tracer air concentrations to within a factor of two of the measured values 20% to 50% of the time, and within a factor of five of the measurements 35% to 85% of the time depending on the complexity of the meteorology and terrain, and the release height of the tracer. These results are achieved without "tuning" the model to any given site, type of source, or sampling method.

A consensus has emerged as to the degree of model sensitivity toward certain important input variables. The key input data and parameters affecting the quality of MATHEW/ADPIC results, in order of decreasing importance, are:

Wind direction. A difference of 10 degrees in wind direction can have a substantial effect on the agreement of model concentrations with individual sampler measurements. This is especially true on small horizontal scales (e.g., 10 km). Errors in model calculations are commonly caused by the lack of a sufficiently representative meteorological (wind) measurements over the model domain. Also, we must contend with the fact that the NOAA and FAA only report wind directions to the nearest ten degrees.

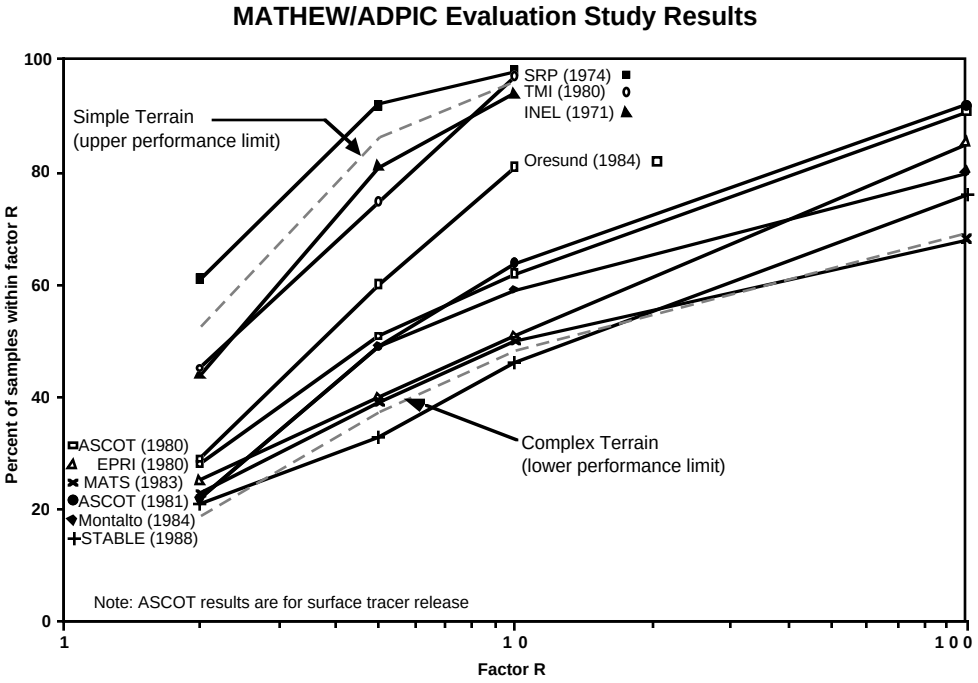
Topography. There is close interaction between topography and the wind field, especially near the surface. Because of the limitations on the number of computational cells available, the resolution of important topographical features, such as riverbeds, ridges, and canyons, is frequently poor. This is most severe in complex terrain situations and under stable atmospheric conditions.

Turbulence parameters. Most turbulence parameters are semi-empirical, even for homogeneous turbulence. To describe the variation of turbulence parameters in space and time for all but the most trivial atmospheric conditions is very difficult. Factors of two to five in concentration

differences computed by MATHEW/ADPIC can appear because of uncertainties in turbulence parameters.

Wind speed. While much less likely to produce errors than the wind direction, the wind speed can, nevertheless, produce errors of factors of two quite easily. Model errors due to wind speed are harder to detect because they are difficult to separate from errors caused by the turbulence parameterization.

Source strength. In controlled experiments the source strength is generally well known; however, in accident situations this is not always true. During an emergency response to an accident, the uncertainty of the source term and the properties of the source material can introduce significant errors into the model results.



STUDY	YEAR	LOCATION	TOPOGRAPHY	ATMOSPHERIC STABILITY	RELEASE HEIGHT
INEL	1971	Idaho Falls, ID	Flat-Rolling	Unstable	Surface
SRP	1974	Aiken, SC	Flat-Rolling	All	Surface, Elevated
ASCOT	1980	The Geysers, CA	Complex	Stable	Surface, Elevated
ASCOT	1981	The Geysers, CA	Complex	Stable	Surface, Elevated
TMI	1980	Harrisburg, PA	Flat-Rolling	All	Elevated
EPRI	1980-1	Kincaid, IL	Flat	All	Elev + Plumerise
∅resund	1984	Oresund Strait, Denmark/Sweden	Flat-Rolling	Neutral (spatially transitional)	Elevated
MATS	1983	Aiken, SC	Flat-Rolling	Neutral, Unstable	Elevated
Montalto	1984	Italy	Flat-Rolling	Unstable	Surface, Elevated
STABLE	1988	Aiken, SC	Flat-Rolling	Stable	Elevated

Fig. II.D.1. Percentage of cases in which MATHEW/ADPIC results are within a factor N of field data.

III. User's Guide

III.A. ARAC Models File System and Flowcharts

Two classes of information need to be passed along the ARAC model sequence. First, the major results of each model must be passed to the next model, mainly as various arrays of data (e.g., model elevations, winds, concentrations). Second, parameters used by one model that are needed by subsequent models for purposes of processing and labeling. There is also a need to maintain a record of the information used in a given model execution to support precise analysis and debugging, i.e., a traceback capability is necessary. The ARAC models file system provides for these three requirements. This section will discuss the file system (both naming conventions and content) which allows the ARAC models to handle these three functions.

The first component of the file system is the (invisible) TraceBack Block (TBB) that appears at the beginning of every file used to pass data between models. It consists of directory and file names that indicate the data files used to generate the associated model results contained in the remainder of the file. For this traceback function to work, each file referenced by the TBB must be uniquely named. These unique names include qualifiers and sequence numbers as described later. The input files indicated may also be read by subsequent models as the source of any parameters required for that model's execution. Thus the only data transferred via binary intermodel files (BIFs) are actual model results. The exception is grid/topography data produced by TOPOG. This information will be read by all the models to define the grid geometry and so occupies a unique place with respect to the rest of the models. The grid definition file (TOPOG*.GRID) contains the grid parameters as well as the model topography cell heights (here we use a convention convenient in discussing these uniquely qualified files: TOPOG*.GRID means all qualified files beginning with TOPOG and ending with the suffix .GRID. Similarly, *.NML means all files which end with the suffix .NML; in addition to files that are required to be uniquely names to support traceback, some other types of files also conform to this convention for consistency, e.g., *.CGMB files). Due to its central place in the model scheme and its small size the TOPOG*.GRID file is maintained in ASCII and can be viewed like any other such file.

Each model is controlled by a namelist file (*.NML) which contains the parameters controlling the model run. These files are read by the appropriate model from generically named files (e.g., TOPOG.NML, MEDIC.NML) after which the information is rewritten to a uniquely named completed namelist file (*.CNML) that will contain the values of all variables in the namelist (including data-loaded defaults) in a namelist-readable format. Subsequent models read these CNML files as needed. Almost all output files contain one data set per file. That exception is the *.LOG files which are intended to record a log of a particular model run and not remain as part of the files documenting the run. *.LOG files have version numbers to distinguish them, but one the ARAC VMS systems multiple files of the same name and a different version number are purged on a frequent basis. Thus, uniquely named files survive a system initiated purge while *.LOG files do not. A single ADPIC run generally processes a number of MEDVEL/MATVEL files. To document this, a file listing

the MEDVEL/ MATVEL files used during the creation of a specific concentration output file (CONC*.BIF) in generated (CONC*.VLST) to pair with it. Similarly, PLCNT and TIMEHIS write files (*.CLST) which documents the CONC files used in the program execution. These lists of data files provide a starting point in the event a complete traceback of an ADPIC, PLCNT or TIMEHIS run should be necessary.

There are three main components that comprise the technical details of this system: 1) file naming conventions, 2) the TBB structure, and 3) a TBB definition capability that allows substantial flexibility in permitting essentially unbounded expansion of the models in terms of the number of files read and written. The descriptions of the last two of these components are implementation details intended largely for programming staff and are described in Appendix C. The rest of this section discusses the file naming conventions.

As mentioned above, the traceback system relies on unique file names for all files maintained as part of it. The need for unique file names and some descriptive information on the file's contents have been combined to create human-readable file names. To assist understanding and file manipulations, the filenames follow one of three standard formats distinguished by the qualifier field. The various fields in a name of a given format type will tend to line up when listed with a directory search command. This feature helps both man and machine to see 'at a glance' what each file contains. The general file name structure has the following stencil:

prefix	qualifier	sequence	extension
7	24	4	4/5

The *prefix* in combination with the extension indicates specific file type. If the file extension is sufficient to identify the file, the name of the model that created the file is used as the prefix. Graphics output files are of this type, for the suffix .CGMB uniquely identifies the file type and the originating program's name is needed for graphics processing.

The *qualifier* identifies the data set as uniquely as possible given the space constraints of the filename. There are three distinct qualifier formats. The first is used with the TOPOG*.GRID files, which contain grid information. The second format includes time information appropriate to the file and applies to most other files in the system. These qualifiers include a starting date and time along with duration or averaging time as appropriate. For clarity, three character alphabetic months are used in dates. The last format uses the first 12 characters of both the problem and run names to qualify the file name.This last type is useful for STNLOC.CMET, where the selection of station locations being output for reference is not dependent on time, but rather the location or region used in the problem. Thus, the problem and run names are sufficient to be unique qualifiers.

GRID files: rrrrrkmzzzzzmxxxxx_yyyyy

rrrrr	=	the larger of east/north range in rounded km
zzzzz	=	the vertical grid range in m
xxxxx	=	the x coordinate of the grid center in rounded km
yyyyy	=	the y coordinate of the grid center in rounded km
underscores	=	unassigned characters

date/time files: ddmmmyy_hhmm_ss_ddd_hhmm

dd	=	the date w/ leading zeros
mm	=	the month as three alphabetic characters
yy	=	the year w/ leading zeros
ddd	=	duration in days w/ leading zeros
hh	=	the hour or number of hours of duration w/ leading zeros
mm	=	the minute or number of minutes of duration w/ leading zeros
ss	=	the seconds or number of seconds of duration w/ leading zeros

NOTE: For conc*.bif and conc*.vlst files, the ddd_hhmm field indicates time from start of run while for most other files that field gives an averaging time or valid time interval.

time-independent files: ppppppppppppprrrrrrrrrrrr

ppppppppppp	=	the first 12 characters of the problem name
rrrrrrrrrrr	=	the first 12 characters of the run name
underscores	=	unassigned characters

Having a *sequence* number which is always changing (increasing) forces unique file names with repeated model executions with the same or similar data. It is always one larger than the highest sequence number of any previously existing file. The sequence number is normally displayed as three digits with a leading underscore for clarity; however, if the highest sequence number of an existing file equals or exceeds 999 then the underscore prefix is usurped to allow sequence numbers up to 9999.

The *extension* indicates the class of files as follows:

.NML	Input namelist file
.CNML	Completed namelist file
.MET	Metdata input file (observations or locations)
.CMET	Completed metdata file (observations or locations)
.DATA	Sampler data input file
.CDAT	Completed sampler data file
.BIF	Binary Intermodel file
.CGMB	Computer Graphics Metafile binary format
.GRID	Model grid file
.VLST	Velocity filename list
.CLST	Concentration filename list
.TBB	PLCNT/TIMEHIS ASCII traceback block
.LOG	Model run log file

The following is a list of the files to be used by the main model system, grouped by the relevent model:

TOPOG.NML
TOPOG__rrrrrkmzzzzzmxxxxx_yyyyynnn.CNML
TOPOG__rrrrrkmzzzzzmxxxxx_yyyyynnn.CGMB
TOPOG__rrrrr kmzzzz mxxxxxyyyyy_nnn .GRID
TOPOG.LOG

MEDIC.NML
MEDIC__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CNML
STNLOC.MET
STNLOC_pppppppppppprrrrrrrrrrr.CMET
OBSERV.MET
OBSERV_ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CMET
MEDIC__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CGMB
MEDVEL_ddmmmyy_hhmm_ss_ddd_hhmm_nnn .BIF
MEDIC.LOG
MEDIC.VLST

MATHEW.NML
MATHEW__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CNML
MATHEW_ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CGMB
MATVEL_ddmmmyy_hhmm_ss_ddd_hhmm_nnn .BIF
MATHEW.LOG
MATHEW.VLST

ADPIC.NML
ADPIC__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CNML
ADPIC__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .RST
ADPIC__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CGMB
CONC___ddmmmyy_hhmm_ss_ddd_hhmm_nnn .BIF
CONC___ddmmmyy_hhmm_ss_ddd_hhmm_nnn .VLST
CONC___ddmmmyy_hhmm_ss_ddd_hhmm_nnn .RST
PARTPOS___ddmmmyy_hhmm_ss_ddd_hhmm_nnn .BIF
ADPIC.LOG
ADPIC.CLST

CONC_ANALYSIS.NML
PLCNT__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CNML
PLCNT__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CGMB
PLCNT__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .CLST
PLCNT__ddmmmyy_hhmm_ss_ddd_hhmm_nnn .TBB
PLCNT.LOG

TIMEHISddmmmyy_hhmm_ss_ddd_hhmm_nnn .CNML
SAMPLER.DATA
SAMPLERddmmmyy_hhmm_ss_ddd_hhmm_nnn .CDAT
TIMEHISddmmmyy_hhmm_ss_ddd_hhmm_nnn .CGMB
TIMEHISddmmmyy_hhmm_ss_ddd_hhmm_nnn .CLST
TIMEHISddmmmyy_hhmm_ss_ddd_hhmm_nnn .TBB
TIMEHIS.LOG

Figure III.A.1 illustrates the flow of the overall CG-MATHEW/ADPIC model system. The step-by-step progression through each of the five executable programs is shown by bolded arrows. (The TIMEHIS program is only used in conjunction with sampler measurement data.) Figures III.A.2 - III.A.4 show in greater detail the files associated with each of the five programs.

The ARAC models can be run on a variety of map projections, which can be selected to minimize the projection-induced scale distortion or to match a projection chosen by an external organization. In addition, to match the details of the geometry of a particular study area it is possible to reposition the origin of a projection as well as to rotate the projection. The models are executed on the current map projection as defined by the logical MAP$F_PROJECTION_PARAMETERS. This logical points at an ASCII file that defines a projection.

Not all map projections are appropriate for modeling with the ARAC models. The applicable projections are conformal, which implies that that a single map scale applies at a given location and that relative direction is unaffected by the projection. The conformal projections currently implemented and available for use in the modeling system are the UTM, Ordnance Survey and GWC North and South specific projections and any normal Mercator, transverse Mercator, Lambert conformal conic, polar stereographic or general stereographic projection. If a model is run with a projection, the wind directions and speeds are corrected in MEDIC to account for the projection-induced distortion. The projection information is also used in the model graphics to indicate the grid orientation via a north arrow or a graticule. As a result, all the models need to know the current projection. The choice of whether to use a projection or not is made in TOPOG with the namelist variable MAP_ADJUSTMENT. If a map projection is not used the winds are not adjusted in MEDIC and no grid orientation indication is made on the graphics.

The step of choosing a map projection takes place in production and Beta when a new problem is created using the Select_Map_Projection and Create_Map_Projection commands. The map projection file is placed in a problem-specific directory that is referenced with the logical MAP$A_PROJECTION_PARAMETERS. Outside of the operational system, the user must ensure that the MAP$F_PROJECTION_PARAMETERS and MAP$A_PROJECTION_PARAMETERS logicals are set to point at the appropriate projection definition file. for additional details, please see the document "The ARAC Map Projection System".

The input files TOPOG.NML, MEDIC.NML, MATHEW.NML, ADPIC.NML, and CONC_ANALYSIS.NML contain model parameters in Fortran namelist format. These namelist input files are created and modified by the user. Each executable produces a log file (primarily a journal file used for debugging purposes) and a graphics file, as well as other files which are described in this section.

TOPOG uses model parameters from TOPOG.NML and terrain data to create the underlying model grid cell heights, which are written to the GRID file.

MEDIC uses model parameters from MEDIC.NML to select meteorological data from various sources and to specify parameters used in interpolating and extrapolating that data. MEDIC can

access met station location either from a Problem Station Library or from a STNLOC.MET namelist file, and meteorological data can be accessed either from Problem Metdata Files or from an OBSERV.MET file. The former methods for met station information and meteorological data are the primary methods used by ARAC operational responses, while the latter methods are used primarily for canned metdata cases. MEDIC produces MEDVEL files, one for each met dataset, containing the three-dimensional gridded wind fields.

CG-MATHEW uses the model parameters from the MATHEW.NML file to adjust the gridded wind fields from the MEDVEL files to make them mass-consistent. These mass-adjusted three-dimensional wind fields are written to MATVEL files.

ADPIC uses the model parameters from the ADPIC.NML file to release marker particles which simulate the actual release of effluent into the wind field. The grid cell concentrations for specified heights for specified plot types for a specified time interval are written to CONC*.BIF files, corresponding CONC*.VLST files are also written containing the MEDVEL/MATVEL filenames used in generating the CONC*.BIF file. Information for individual particles can be saved in the ADPIC*.RST file, so that an ADPIC run can be restarted. Information required for displaying dots over terrain is contained in PARTPOS*.BIF files.

PLCNT and TIMEHIS are post-processor codes which use the parameters from the CONC_ANALYSIS.NML file to display the concentration data from the CONC*.BIF files in different ways: PLCNT generates contour plots of the concentration data, while TIMEHIS generates semi-log time history plots of the concentration data.

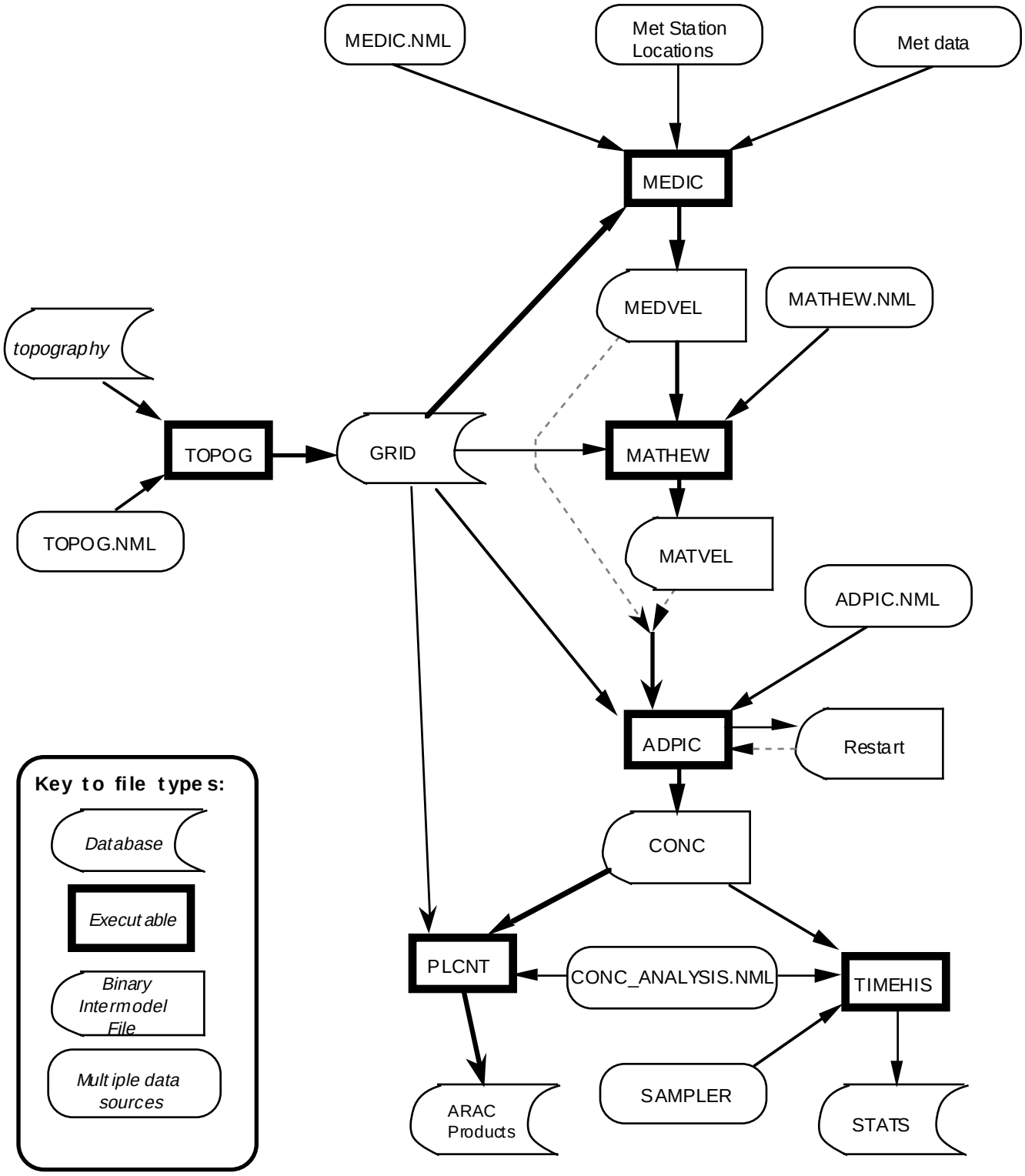


Fig. III.A.1 Overview of the CG-MATHEW/ADPIC Models

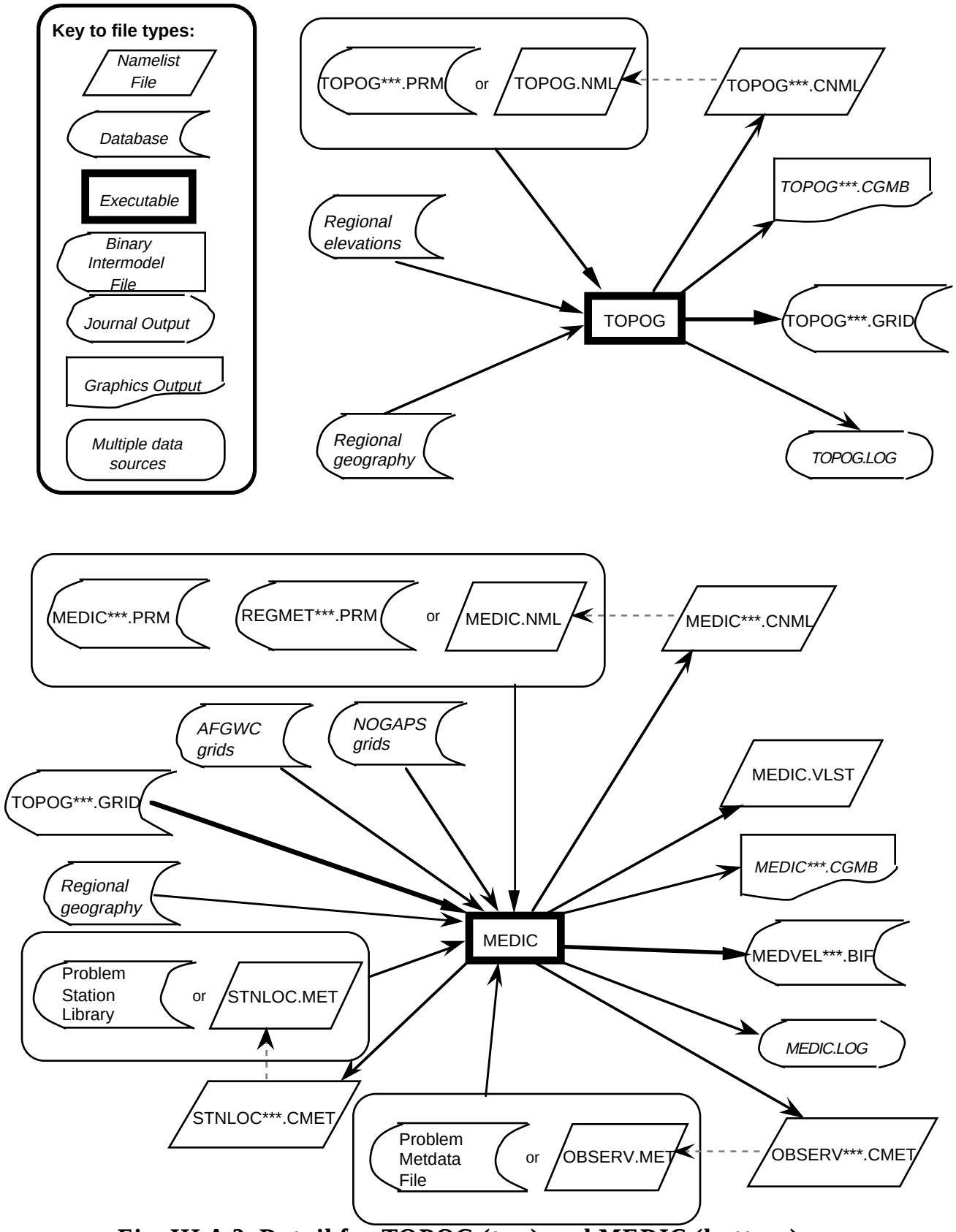


Fig. III.A.2 Detail for TOPOG (top) and MEDIC (bottom)

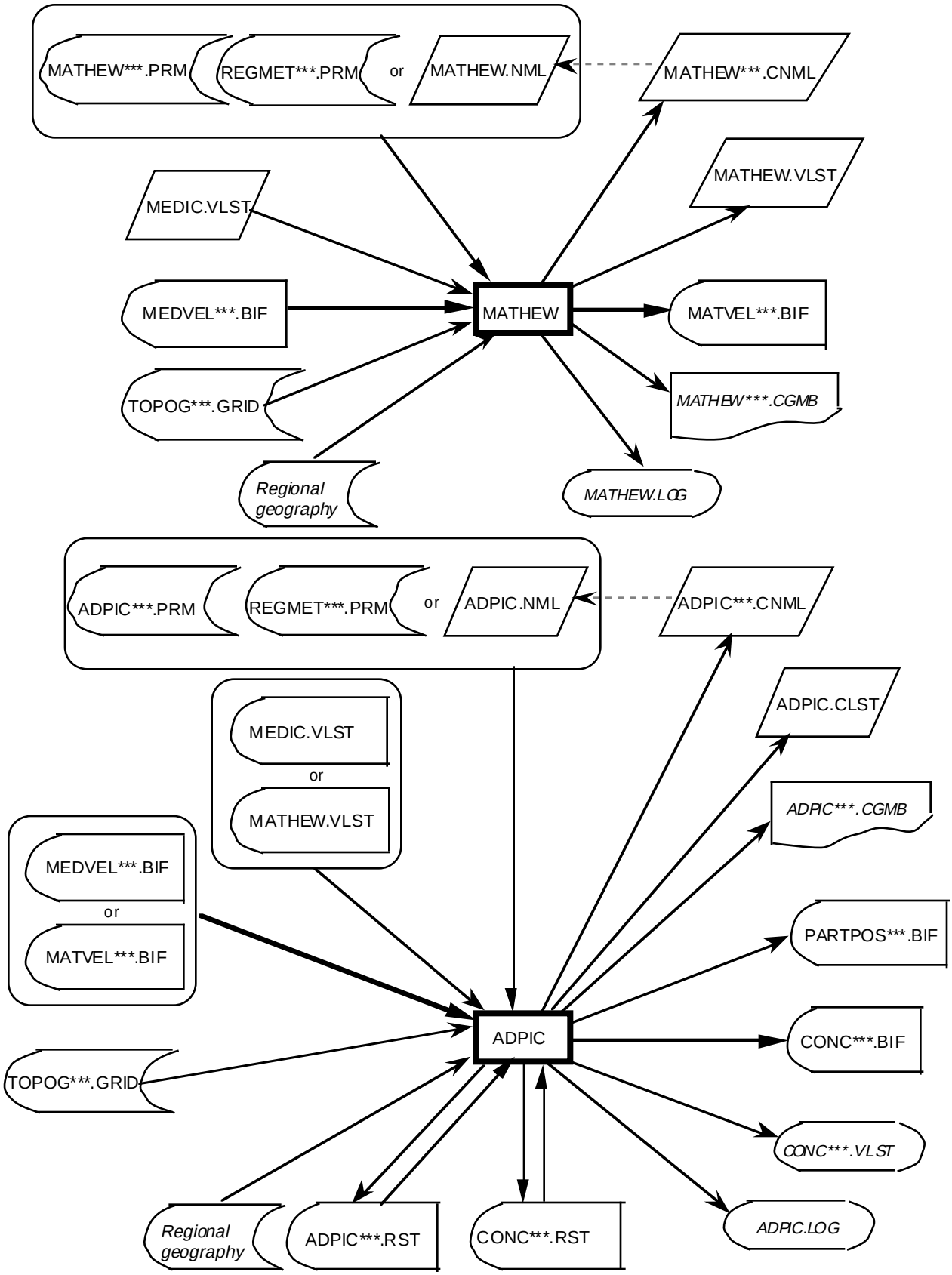


Fig. III.A.3 Detail for CG-MATHEW (top) and ADPIC (bottom)

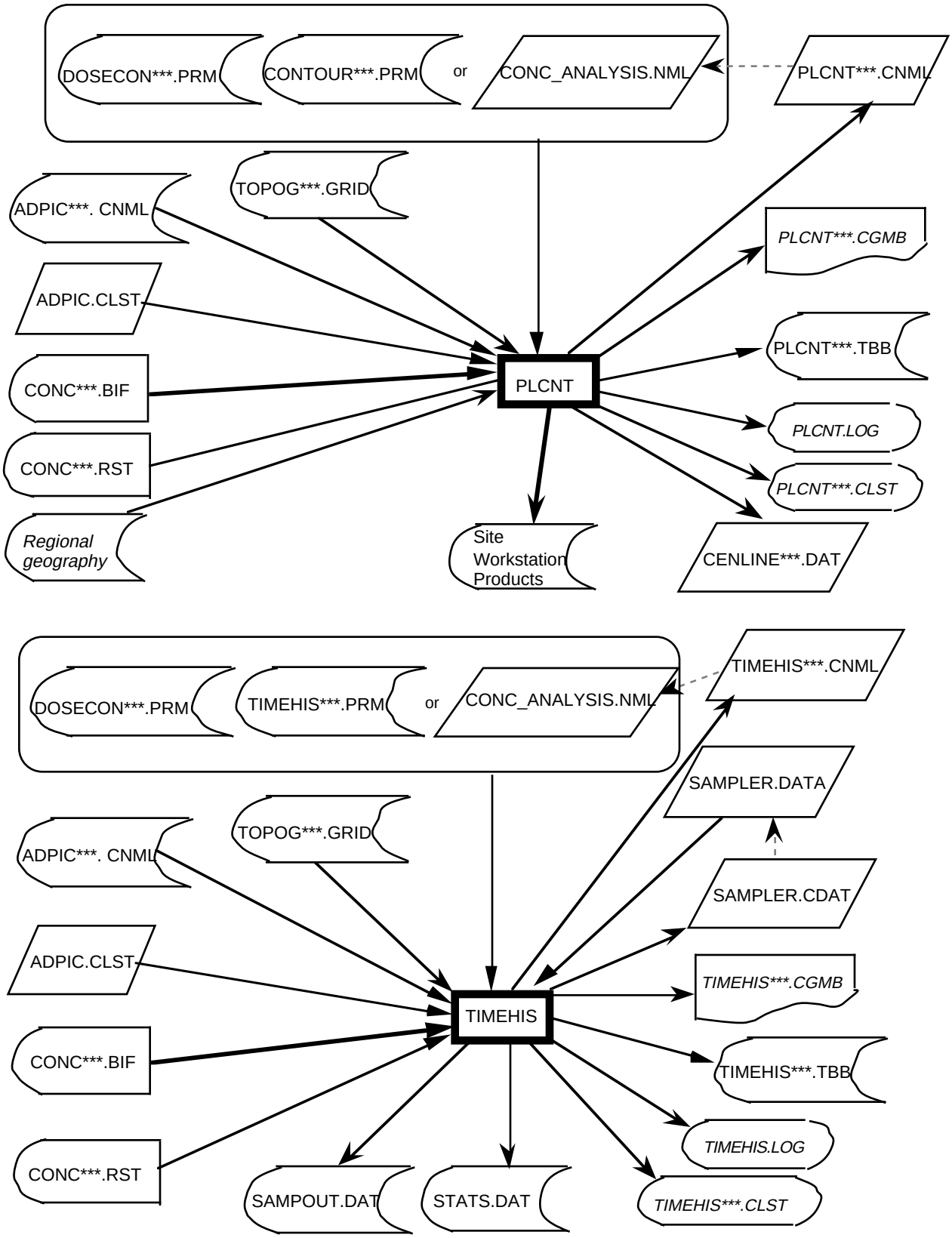


Fig. III.A.4 Detail for PLCNT (top) and TIMEHIS (bottom)

III.B. Input Files

There are several conventions that are followed in all the model input files. All program control flags are input as single characters with acceptable values of 'Y', 'y', 'N', and 'n'. Horizontal distances are input in kilometers and vertical distances in meters. Dates are all input as seven character strings in ddMMMyy format where dd is the day of the month, MMM is the alphabetic month (case insensitive) and yy is the last two digits of the year. The date/time utilities in the models work from January 1, 1950 to December 31, 2049 by assuming the yy values of less than 50 are in the next century and yy values of greater than 50 are in this century. Times are strings of up to eight characters and can be in one of the four following formats:

'hhmm'

'hhmmss'

'hh:mm'

'hh:mm:ss'

where hh is the hours (<24), mm is the minutes (<60) and ss is an optional seconds field (<60). Note the in the description of the time input variables below, they will typically be described as being in 'hhmmss' format although all of the formats above are acceptable. There are also a number of time interval inputs (e.g., ..._INTERVAL variables in ADPIC and ..._STEP variables in PLCNT) where a prefix can be added to a legal time string to indicate a number of days (<1000) in the duration. Thus, the following is a legal interval input string (note that the days field along with the double colon are not required if the interval is less than 24 hours but if the days field is included then the double colon must separate the days and hours fields):

'DDD::hh:mm:ss'

III.B1. TOPOG.NML

TOPOG.NML is the input file of grid size and position information for the TOPOG code. Since TOPOG creates a grid that is fixed in space, a large enough area must be chosen to account for the wind and stability conditions that might affect the effluent as the problem unfolds. An overview of the input is necessary because of the appreciable interplay between the individual parameters in TOPOG.NML as will as other parameters in the code system.

Since the TOPOG.NML input defines a three-dimensional grid for ADPIC, the first step in preparing the file is to estimate the area of interest for the duration of the problem. The overall objective is to keep at least 90% of the particles within the grid volume so as to maximize resolution of the effluent puff or plume. We have found streamline analyses using past, present and, if available, forecast weather maps to be extremely helpful in estimating the effluent trajectory. A calculated mean wind speed in conjunction with this directional information will then serve to guide the analyst in choosing the proper horizontal (x and y) grid dimensions. Care should be taken to

insure that the resulting ADPIC grid will have positive x-y coordinates, since PLCNT will fail in low level graphics calls with negative coordinates.

In selecting the vertical grid dimension, it is necessary to consider the height of the release, topography, upper air winds, and the stability of the atmosphere. Topography is one of the more important considerations in this decision process. TOPOG assigns a zero height to the lowest terrain elevation in the grid. The top of the grid is generally chosen to be above the highest topographic feature. There are two exceptions to this rule: (1) if an isolated mountain peak extends well above the nearby topography, the peak should be allowed to protrude through the top of the grid so as to improve the resolution elsewhere, and (2) if it is highly unlikely that the effluent cloud would rise above a particular terrain level, it may be advantageous to lower the grid top to that level.

There are two ways of altering the grid dimension: (1) the size of the grid cell can be varied in the TOPOG.NML file as described below, and (2) the number of grid points along each axis can be varied. (For example the R41 see appendix F, CG-MATHEW and ADPIC grids have 51 by 51 by 15 points and 41 by 41 by 15 points, respectively.) The first option is the one of choice in most ARAC problems. However, there may be times when it is desirable to change the number of grid points to achieve the optimum grid resolution for a given grid volume. In order to do this, the grid point parameters must be changed in appropriate include files and the codes recompiled. The parameter names in TOPOG, MEDIC, and CG-MATHEW and ADPIC are IMAXPIC, JMAXPIC, and KMAXPIC for the ADPIC grid and IMAXMAT, JMAXMAT, and KMAXMAT for the CG-MATHEW grid. (Computer memory and execution speeds are limiting factors in choosing a total number of grid points.)

The following is a list of the TOPOG.NML variables grouped alphabetically within namelist blocks:

II.B1a. TOPOG_CONTROL namelist

CONTOUR_INTERVAL

The model terrain contour interval in meters.

(Default: CONTOUR_INTERVAL = DELZ)

Warning errors:	CONTOUR_INTERVAL $\leq$ 1.0
Fatal errors:	CONTOUR_INTERVAL $\leq$ 0.001
	CONTOUR_INTERVAL > 5000.0 (M/A)

FLAT_TOPO

Flag ('Y' or 'N') indicating if flat terrain is to be used. Special Note for HM/A: valid values for the terrain file (defined by the “Select Topography” menu option) are: NHEMISTD, NHEMIAVG, NHEMIMAX.

(Default: FLAT_TOPO = 'N' $\rightarrow$ do not use flat terrain)

INTERACTIVE

Flag ('Y' or 'N') specifying whether or not user interaction is required during the running of TOPOG. Minimal user interaction involves resolution of any 1* DELX "holes"; in the noninteractive mode, the holes are filled automatically.

(Default for HANDSOFF mode: INTERACTIVE = 'N' → no user interaction; Default for non-HANDSOFF mode: INTERACTIVE = 'Y' → user interaction required)

LISTDETAILS

Flag ('Y' or 'N') specifying whether or not additional information about the TOPOG grid will be written to the echo file. The additional information consists of: cell heights and DELZ heights.

(Default: LISTDETAILS = 'N' → no additional information is written to the echo file)

RESTART

Flag ('Y' or 'N') indicating that an existing TOPOG*.GRID file is to be opened and read. This allows a previous grid to be modified interactively without rerunning all of TOPOG.

(Default: RESTART = 'N' → start from scratch and create new TOPOG*.GRID)

III.B.1.b. TOPOG_PARAMS namelist

DELX

The grid cell size in kilometers along the *x*-axis. DELX must be an integer multiple of one-quarter the terrain database resolution. For example, an ARAC elevation file with a resolution of 0.5km, the DELX must be a multiple of 0.125km. DELX should be 381.0 for an H51 grid and 190.5 for an H101 grid (see Appendix F for operational model grid conventions).

Warning errors:	none
Fatal errors:	DELX $\leq$0.001
	DELX >1250.0

DELY

The grid cell size in kilometers along the *y*-axis. DELY must be an integer multiple of one-quarter the terrain database resolution. For example, an ARAC elevation file with a resolution of 0.5km, the DELX must be a multiple of 0.125km. DELY should be 381.0 for DELX should be 381.0 for an H51 grid and 190.5 for an H101 grid (see Appendix F for operational model grid conventions).

Warning errors:	none
Fatal errors:	DELX $\leq$0.001
	DELX >1250.0

DELZ

The grid cell size in meters along the *z*-axis. See Appendix F for the number or vertical grid cells in the operationally supported grids.

Warning errors:	DELZ > 1000.0
	DELX <1.0
Fatal errors:	DELZ ≤0.001
	DELZ > 5000.0

DOUBLECELL

Flag ('Y' or 'N') specifying whether or not grid cell heights will be assigned in blocks of four grid cells from the lower left corner of the grid.

(Default: DOUBLECELL = 'N' → blocks of four will not be used, i.e., the height for each grid cell is determined individually.)

MAP_ADJUSTMENT

Flag ('Y','N') specifying whether or not a map projection adjustment is to be made which would align map north with true north for the map projection selected. One effect of setting this flag is to correct the wind direction arrays, and adjust the wind speed to account for local scale variations caused by the map projection. Note: In order to use MAP_ADJUSTMENT ='Y', the map projection logicals need to be set, i.e., you may need to have selected a map projection.

(Default: MAP_ADJUSTMENT = 'Y')

RANGE

ADPIC grid extent along both the x- and y axes in kilometers. RANGE can be used to specify both XRANGE and YRANGE. RANGE may also be used instead of DELX—also see notes for DELX.

Warning errors:	RANGE < 1.0
Fatal errors:	RANGE ≤0.5
	RANGE > 45000.0

X_GRIDCENTER

The model x coordinate of the center of the ADPIC grid, input in kilometers.

Note:	X_GRIDCENTER and Y_GRIDCENTER may be used instead of X_ORIGIN_PIC and Y_ORIGIN_PIC.
	*Special note for H51 and H101 X_GRIDCENTER = 8763 should be used.
Fatal errors:	X_GRIDCENTER ≤ -45000.0
	X_GRIDCENTER > 45000.0

X_OFFSET_MAT

Real number specifying the offset (km) in the x-direction of the ADPIC grid from the CG-MATHEW grid. The ADPIC grid must be contained in the CG-MATHEW grid; therefore X_OFFSET_MAT must be less than 0.5 * (CG-MATHEW grid size - ADPIC grid size).

(Default: X_OFFSET_MAT = 0.0 → no offset)

X_ORIGIN_PIC

The Model x coordinate of the lower left-hand corner of the ADPIC grid in kilometers.

Note: X_ORIGIN_PIC and Y_ORIGIN_PIC may be used instead of X_GRIDCENTER and Y_GRIDCENTER.

Fatal errors: X_ORIGIN_PIC ≤ -45000.0
X_ORIGIN_PIC > 45000.0

XRANGE

ADPIC grid extent along the x-axis in kilometers. XRANGE may be used instead of DELX—also see notes for DELX.

Warning errors: XRANGE < 1.0

Fatal errors: XRANGE ≤0.5
XRANGE > 45000.0

Y_GRIDCENTER

The model y coordinate of the center of the ADPIC grid, input in kilometers.

Note: X_GRIDCENTER and Y_GRIDCENTER may be used instead of X_ORIGIN_PIC and Y_ORIGIN_PIC.
Special note for H51 and H101: Y_GRIDCENTER = 9525.,should be used.
Fatal errors: Y_GRIDCENTER ≤ -45000.0
Y_GRIDCENTER > 45000.0

Y_OFFSET_MAT

Real number specifying the offset (km) in the y-direction of the ADPIC grid from the CG-MATHEW grid. The ADPIC grid must be contained in the CG-MATHEW grid; therefore Y_OFFSET_MAT must be less than 0.5 * (CG-MATHEW grid size - ADPIC grid size).

(Default: Y_OFFSET_MAT = 0.0 → no offset)

Y_ORIGIN_PIC

The Model y coordinate of the lower left-hand corner of the ADPIC grid in kilometers.

Note: X_ORIGIN_PIC and Y_ORIGIN_PIC may be used instead of X_GRIDCENTER and Y_GRIDCENTER.

Fatal errors: Y_ORIGIN_PIC ≤ -45000.0
Y_ORIGIN_PIC > 45000.0

YRANGE

ADPIC grid extent along the y-axis in kilometers. YRANGE may be used instead of DELY–also see notes for DELY.

Warning errors: YRANGE < 1.0

Fatal errors: YRANGE ≤0.5

YRANGE > 45000.0

ZRANGE

ADPIC grid extent along the z-axis in meters. ZRANGE may be used instead of DELZ – also see notes for DELZ.

Warning errors: ZRANGE < 10.0 (M/A)

ZRANGE ≥5000.0

Fatal errors: ZRANGE > 70000.0 (M/A)

Sample TOPOG.NML Input File (with center reference)

```
$topin_control                          beginning of control namelist
FLAT_TOPO          = 'Y'                flag for using flat terrain
RESTART            = 'N'                flag for using existing GRIDIN file
INTERACTIVE        = 'Y'                flag for non-interactive execution
LISTDETAILS        = 'N'                flag for writing additional information to echo file
$end                                    end of control namelist

$topin_params                           beginning of params namelist
X_GRIDCENTER       = 115.0      ←       X of center of ADPIC model grid
Y_GRIDCENTER       = 115.0      ←       Y of center of ADPIC model grid
RANGE              = 30.000     ←       range (km) of ADPIC grid in x- and y-directions
ZRANGE             = 700.       ←       range (m) of ADPIC grid in z-direction
X_OFFSET_MAT       = 0.000      ←       offset (km) for CG-MATHEW grid in x-direction
Y_OFFSET_MAT       = 0.000      ←       offset (km) for CG-MATHEW grid in y-direction
DOUBLECELL         = 'N'        ←       flag for assigning grid cell heights in blocks of four
$end                                    end of params namelist
```

Sample TOPOG.NML Input File (with lower left corner reference)

```
$topin_control                          beginning of control namelist
FLAT_TOPO          = 'Y'                flag for using flat terrain
RESTART            = 'N'                flag for using existing GRIDIN file
INTERACTIVE        = 'Y'                flag for non-interactive execution
LISTDETAILS        =' N'                flag for writing additional information to echo file
$end                                    end of control namelist

$topin_params                           beginning of params namelist
X_ORIGIN_PIC       = 100.0      ←       X of lower left corner of ADPIC model grid
Y_ORIGIN_PIC       = 100.0      ←       Y of lower left corner of ADPIC model grid
RANGE              = 30.000     ←       range (km) of ADPIC grid in x- and y-directions
ZRANGE             = 700.       ←       range (m) of ADPIC grid in z-direction
X_OFFSET_MAT       = 0.000      ←       offset (km) for CG-MATHEW grid in x-direction
Y_OFFSET_MAT       = 0.000      ←       offset (km) for CG-MATHEW grid in y-direction
DOUBLECELL         = 'N"        ←       flag for assigning grid cell heights in blocks of four
$end                                    end of params namelist
```

III.B.2. Geography

The geography file contains location and classification information about geographic features (e.g. roads, lakes, rivers, political boundaries). This information is used to plot a base map for some of the model graphics products. These files are normally not provided with the models.

III.B.3. Problem Elevation Data

The problem elevation file contains the altitudes of the topographic surface around a site. It is composed of a header that describes the grid being used, followed by a two-dimensional array of cell heights, one for each grid cell. These files are in netCDF format. netCDF files are binary data files together with public-domain software that should be installed on all systems that run the ARAC models. The software together with the data format form a system-independent means of storing and transfering data between programs. TOPOG reads these netCDF elevation files (by linking against netCDF library software), which are named according to the convention:

sssssssssmppppppppp_rrrrr_sss.ELEV:

where "sssssssss" is the site_id or equivalent, "m" is an underscore for unmodified projections or an 'M' for modified projections, "pppppppp" is a map projection string, e.g.,

'UTM_____'

'OS_____'

'GWCNORTH_'

'GWCSOUTH_'

'GWCTROPIC'

'NMERCATOR'

'TMERCATOR'

'LAMBERTCC'

'POLARSTER'

"rrrrr" is the map range in km and "sss" is a sequence number that ensures that the filename will be unique.

Elevation File Header

The header contains the following information:

NetCDF problem elevation file header:

elev_filename	name of this elevation data file as described above
elev_site_id	the site_id associated with this map
elev_description,	textual description of nature of geodata
elev_method_string	textual description of processing method (stored as global attribute)
elev_grid_string	textual description of grid structure (stored as global attribute)
elev_data_source_1	primary source of the elevation data used to generate this file,e.g.,'DMA Planar'; 'DMA DTED'; 'ARAC 500m'; 'ETOPO5'
elev_data_source_2	secondary source of the elevation data used to generate this file, a secondary source is mainly required to fill holes in the primary data coverage
elev_projection_filename	filename of projection definition file
elev_projection_name	string indicating the map projection (ARAC std); see projection_names.cli in geogutil
elev_projection_id	index indicating the map projection (ARAC std); see projection_names.cli in geogutil
elev_ellipsoid_name	string indicating the earth figure (ellipsoid or sphere) (USGS GCCP std); see projection_names.cli in geogutil
elev_ellipsoid_id	index indicating the earth figure (ellipsoid or sphere) (USGS GCCP std); see projection_names.cli in geogutil
elev_central_latitude	central latitude of map projection for this elevation file
elev_central_longitude	central longitude of map projection for this elevation file
elev_std_parallel_1	first standard parallel of map projection for this elevation file
elev_std_parallel_2	second standard parallel of map projection for this elevation file
elev_scale_factor	scale factor for the map projection used for this elevation file
elev_false_easting	false easting for the map projection used for this elevation file

elev_false_northing	false northing for the map projection used for this elevation file
elev_origin_latitude	latitude of origin for a modified map projection used for this elevation file
elev_origin_longitude	longitude of origin for a modified map projection used for this elevation file
elev_origin_rotation	rotation of origin for a modified map projection used for this elevation file
elev_datum_name	datum name for this map
xmin_domain_elev	minimum x coordinate of largest normal plotting window (km)
ymin_domain_elev	minimum y coordinate of largest normal plotting window (km)
xmax_domain_elev	maximum x coordinate of largest normal plotting window (km)
ymax_domain_elev	maximum y coordinate of largest normal plotting window (km)
west_limit_elev	minimum longitude coordinate of grid
east_limit_elev	maximum longitude coordinate of grid
south_limit_elev	minimum latitude coordinate of grid
north_limit_elev	maximum latitude coordinate of grid
x_cell_size_elev	grid cell size in the x direction (km)
y_cell_size_elev	grid cell size in the y direction (km)
x_dimension_elev	dimension in the x direction
y_dimension_elev	dimension in the y direction
elev_minimum	minimum elevation in the domain (m)
elev_maximum	maximum elevation in the domain (m)

Elevation Grid

The array of elevations corresponds to the grid defined in the header. Each grid cell is associated with an array index pair (i,j), with the pair (1,1) indicating the grid cell at the southwest

corner of the grid. The value of an array element is usually the average height (although other resampling options are available, see The ARAC II Geodata System for details) in meters above sea level of the corresponding grid cell. An array value of -9999 indicates that no valid data was available to generate an elevation for this grid cell. TOPOG treats these missing data flags as sea level (0) heights. TOPOG does not care what the dimensions of the elevation grid in the netCDF elevation file are but to use the SHOTER (i.e., PERSPEC) function in operations the grids should be left at the default of 400 by 400.

III.B.4. MEDIC.NML

MEDIC.NML is the principal input file for control of the MEDIC code. It provides MEDIC with numerous parameters controlling the wind field extrapolation or interpolation procedures Two types of Fortran namelists make up the MEDIC.NML file. The first, MEDIC_CONTROL, permits the user to control general processing and output options. The second, MEDIC_PARAMS, controls parameters which are specific to a set of meteorological data. A separate MEDVEL output file is created for each occurrence of a MEDIC_PARAMS namelist within MEDIC.NML. These MEDVEL data sets are used by CG-MATHEW to create snapshots of the atmospheric winds adjusted for mass-consistency throughout a gridded volume.

Meteorological data may come from an OBSERV.MET file or Problem Metdata files, and/or GWC or RWM gridded met data files. The variable METDATASET given in a MEDIC_PARAMS namelist is matched to a keyword which prefaces every METDATASET within an OBSERV.MET file. Typically, the MEDIC_PARAMS namelist and the corresponding METDATASET within the OBSERV.MET file track at intervals which match data availability, typically every hour.

For the Vax versions of MEDIC, the source of metdata (Problem Metdata Files, OBSERV.MET file, or gridded data) is not related to the source of met station loc ation information (Problem Station Library or STNLOC.MET file). However, for the Alpha versions of MEDIC, the STNLOCPSL parameter in the MEDIC.NML file controls whether or not the build_ascii_probmet executable will be run by the Start_Medic (STAMED) command: if STNLOCPSL = 'Y', then a STNLOC.MET and OBSERV.MET file are created for an R81 MEDIC run; i.e., If STNLOCPSL = 'N' for an R81 MEDIC run, then you have to already have a STNLOC.MET and an OBSERV.MET file in the run directory for MEDIC to run on the Alpha machine.

The table below describes some of the combinations of initial metdata ("initial" refers to what metdata info exists prior to using the STAMED command) and met station location info, and how these combinations will work for the VAX versions (R41, R51) and ALPHA version (R81) of regional MEDIC. (PMF = Problem Metdata File, and PSL = Problem Station Library)

Stn Location Info Initial Metdata Info	Problem Station Library (STNLOCPSL = 'Y')	STNLOC.MET file (STNLOCPSL = 'N')
PMF's PROBMET_DST... (no OBSERV.MET) METDATASET	VAX: yes, using PMF's _____ ALPHA: yes, using a new OBSERV.MET and STNLOC.MET created from the PMF's and PSL via STAMED (which runs Build_ascii_probmet)	VAX: yes, using PMF's _____ **ALPHA: no,** since PMF's will not be converted
PMF's PROBMET_DST... OBSERV.MET METDATASET	VAX: yes, using PMF's _____ ALPHA: yes, using a new OBSERV.MET and STNLOC.MET created from the PMF's and PSL via STAMED (which runs Build_ascii_probmet)	VAX: yes, using PMF's _____ ALPHA: yes, using existing OBSERV.MET
PMF's PROBMET_DST... (no OBSERV.MET) (no METDATASET)	VAX: yes, using PMF's _____ **ALPHA: no,** since METDATASET not defined	VAX: yes, using PMF's _____ **ALPHA: no,** since PMF's will not be converted
(no PMF's) (no PROBMET_DST...) OBSERV.MET METDATASET	VAX: yes, using existing OBSERV.MET _____ **ALPHA: no,** since no PMF's to convert	VAX: yes, using existing OBSERV.MET _____ ALPHA: yes, using exising OBSERV.MET
(no PMF's) (no PROBMET_DST...) (no OBSERV.MET) (no METDATASET) ALL_WINDS...	VAX: yes, using ALL_WINDS... (doesn't use PSL) _____ **ALPHA: no,** since METDATASET not defined	VAX: yes, using ALL_WINDS... (doesn't use STNLOC.MET) _____ ALPHA: yes, using ALL_WINDS... (doesn't use STNLOC.MET)

PMF's PROBMET_DST... (no OBSERV.MET) METDATASET SINGLE_PROFILE_WINDS	VAX: yes, using single profile from PMF _____ **ALPHA: no,** but it should work; ASRF is in	VAX: yes, using single profile from PMF _____ **ALPHA: no,** sinc PMF's will not be converted

Because CG-MATHEW does a minimal wind field adjustment and because the advection winds in ADPIC persist for extended periods, it is very important that the analyst provide the best possible set of extrapolated winds. The extrapolated winds are sensitive to relatively small changes in the MEDIC parameter values, and the effects on the wind field are often complex and hard to visualize. It is therefore essential that the assessor (a meteorologist) carefully examines the plots of the extrapolated winds which MEDIC produces to determine the validity of the wind field. If the wind field fits the assessor's concept of the state of the atmosphere, then CG-MATHEW may be run with the data set. Otherwise, several of the variables in MEDIC.NML may have to be changed and MEDIC run again. It may also be necessary to omit meteorological data if it is obviously erroneous, e.g., several hours of unchanged direction readings from a station may indicate a "frozen" wind vane. This process is iterative, and is continued until the assessor is satisfied that the best extrapolation (consistent with the data) has been produced. If only gridded upper air data is used, profile extrapolation is generally to be preferred over parameterized extrapolation (PROFILE_EXTRAP = 'Y').

The hemispheric variant HMEDIC (see Part II.C.2.e) uses a subset of the MEDIC.NML namelist variables. All of the MEDIC_CONTROL namelist variables apply *except* for REGMET_PARAMFILE, SFC_ROUGH_HGT, and STNLOCPSL. However, since HMEDIC uses only AFGWC gridded winds and does not perform extrapolation, *only* MET_START_DATE, MET_START_TIME, AVGTIME, NHEDATA, NHEFCST. NHETIME, SHEDATE, SHEFCST, SHETIME are used from the MEDIC_PARAMS namelist.

The following is a list of variables for the two MEDIC namelists. including defaults. Variables not maked as (MEDIC/HMEDIC variable) apply to MEDIC *only*.

III.B.a. MEDIC_CONTROL namelist

GWC_BARB_PLOT_FREQ (MEDIC/HMEDIC variable)

Integer value specifying the density of barbs on the GWC wind barb plots. A value of one (1) indicates that a vector is plotted for every observation. A value of two (2) indicates that a vector is plotted for every second observation, etc.

(Default : 1)

Warning errors: None

Fatal errors: GWC_BARB_PLOT_FREQ < = 0

GWC_BARB_SCALE (MEDIC/HMEDIC variable)

Floating point value scaling the size of the barbs on the GWC wind barb plots in the horizontal direction.

(Defaults : GWC_BARB_SCALE = 0.35 for 100< # stns <500,

GWC_BARB_SCALE = 0.6 for # stns <100,

GWC_BARB_SCALE = 0.2 for #stns >500)

Warning errors: None

Fatal errors: GWC_BARB_SCALE < = 0.0

HORZ_VECTOR_SCALE (MEDIC/HMEDIC variable)

Floating point value which scales the size of the vectors on the vector plots in the horizontal direction.

(Default : 0.5 for MEDIC, 0.1 for HMEDIC)

Warning errors: None

Fatal errors: HORZ_VECTOR_SCALE < = 0.0

IPROFILE (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard R41 CG-MATHEW grid) representing the *i* indices of the sample profiles to be plotted. The first value in IPROFILE is matched with the first value in JPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for IPROFILE with a standard 51 by 51 by 15 R41 CG-MATHEW grid is: 14 26 38 14 26 38 14 26 38. This gives the normal nine sample profiles. The variables XPROFILE and YPROFILE allow desired locations for vertical profiles to be entered in current map coordinates. See the description of those variables for more information.

JPROFILE (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard R41 CG-MATHEW grid) representing the *j* indices of the sample profiles to be plotted. The first value in JPROFILE is matched with the first value in IPROFILE to give the horizontal grid indices of the first profile. Succeeding values from each list are matched in a similar fashion. The default for JPROFILE with a standard 51 by 51 by 15 R41 CG-MATHEW grid is: 14 14 14 26 26 26 38 38 38. This gives the normal nine sample profiles. The variables XPROFILE and YPROFILE allow desired locations for vertical profiles to be entered in current map coordinates. See the description of those variables for more information.

MEDIC_VLST (MEDIC/HMEDIC variable)

6-character string ('NEW', 'NONE', 'MODIFY') which controls the creation of the MEDIC.VLST file. If set to 'NEW', a MEDIC.VLST file is created containing a list of the fully qualified MEDVEL file names produced by the run. If set to "NONE', no MEDIC.VLST file is created. If the flag is set to 'MODIFY', the existing MEDIC.VLST file is read in and updated from the MEDVEL files created during the current run.

(Default MEDIC_VLST = 'NEW'.)

Warning errors: None

Fatal errors: None

NODDS_BARB_PLOT_FREQ (MEDIC/HMEDIC variable)

Integer value (1,2,?) controlling the barb density: one out of every NODDS_BARB_PLOT_FREQ will be plotted on the NODDS horizontal vector plots.

(Default NODDS_BARB_PLOT_FREQ = 1 -> every barb is plotted)

NODDS_BARB_SCALE (MEDIC/HMEDIC variable)

Floating point value (>0, usually less than 2) that scales the barbs on the NODDS horizontal vector plots.

(Default NODDS_BARB_SCALE = 1 -> default barb size is used)

PLOT_LEVELS (MEDIC/HMEDIC variable)

Array (max size = KMAXMAT = 15) of integers (1 to 15 for standard R41 MEDIC grid) representing the *K* levels above terrain at which horizontal vector plots are to be produced.

(Default PLOT_LEVELS : 1, 2, 7, 15)

PLOT_LEVELS_NHE (MEDIC/HMEDIC variable)

Array (max size = KMAX_UPR_HEMI = 15 currently, the number of hemispheric grid levels) of integers representing the Northern Hemisphere data grid levels to be plotted in the MEDIC graphics file. For example, PLOT_LEVELS_NHE=1 4 15 will plot the first (1000~mb), fourth (500~mb), and fifteenth (10~mb) levels.

(Default PLOT_LEVELS_NHE : 1, 5, 10, 15)

PLOT_LEVELS_NODDS (MEDIC/HMEDIC variable)

Array (max size = KMAX_UPR_NODDS = 9) of integers representing the NODDS (Navy Oceanographic Data Display System) data grid levels (1000mb, 925mb, 850mb, 700mb, 500mb, 400mb, 300mb, 200mb, 100mb) to be plotted as barbs in the MEDIC graphics file. For example, PLOT_LEVELS_NODDS=1 2 3 5 will plot the first (1000mb), second (925mb), third (850mb), and fifth (500mb) levels.

(Default PLOT_LEVELS_NODDS = 0 -> no levels plotted)

PLOT_LEVELS_RWM

Array (max size = KMAX_UPR_RWM = 9 currently, the number of Relocatable Window Model levels) of integers representing the Relocatable Window Model grid levels to be plotted in the MEDIC graphics file. For example, PLOT_LEVELS_RWM = 1 2 6 9 will plot the first (surface), second (1000~ft), sixth (15000~ft), and ninth (30000~ft) levels.

PLOT_LEVELS_SHE (MEDIC/HMEDIC variable)

Array (max size = KMAX_UPR_HEMI = 15 currently, the number of hemispheric grid levels) of integers representing the Southern Hemisphere data grid levels to be plotted in the MEDIC graphics file. For example, PLOT_LEVELS_SHE= 1 4 15 will plot the first (1000~mb), fourth (500~mb), and fifteenth (10~mb) levels.

(Default PLOT_LEVELS_SHE : 1, 2, 7, 15)

PRINT_WINDFIELDS (MEDIC/HMEDIC variable)

Flag ('Y', 'N') that controls if the reference level and upper level winds (MEDIC) or the GWC input winds (HMEDIC) and the extrapolated wind fields are to be printed.

(Default : 'N' -> none of the above wind fields are printed)

Warning errors: None

Fatal errors: None

REGMET_PARAMFILE

22-character string which gives the name of the regmet parameter file to be used. The default value, ' ', is used to check if a string was given in the MEDIC_CONTROL namelist. (WARNING: Not yet implemented! !)

(Default : ' ')

Warning errors: None

Fatal errors: None

RWM_BARB_PLOT_FREQ (MEDIC/HMEDIC variable)

Integer value that controls the density of barbs on the RWM wind barb plots. A value of one (1) indicates that a vector is plotted for every observation. A value of two (2) indicates that a vector is plotted for every second observation, etc.

(Default : 1)

Warning errors: None

Fatal errors: RWM_BARB_PLOT_FREQ < = 0

RWM_BARB_SCALE (MEDIC/HMEDIC variable)

Floating point value that scales the size of the barbs on the RWM wind barb plots in the horizontal direction.

(Default : set according to the number of barbs plotted)

Warning errors: None
Fatal errors: RWM_BARB_SCALE < = 0.0

SFC_BARB_PLOT_FREQ

Integer value that controls the density of barbs on the surface wind barb plots. A value of one (1) indicates that a vector is plotted for every surface observation. A value of two (2) indicates that a vector is plotted for every second surface observation, etc.

(Default : 1)

Warning errors: None
Fatal errors: SFC_BARB_PLOT_FREQ < = 0

SFC_BARB_SCALE

Floating point value that scales the size of the barbs on the surface wind barb plots in the horizontal direction.

(Default : set according to the number of barbs plotted)

Warning errors: None
Fatal errors: SFC_BARB_SCALE < = 0.0

SFC_ROUGH_HGT

Floating point value that represents the surface roughness height in meters. The surface roughness height is approximately the average height of most land-based features (e.g., grass, trees, buildings) divided by 30. The analyst should consider features over the entire gridded area before entering a value. This is a subjective parameter since we are usually dealing with very diverse geographical areas. Figure III.B.3 gives typical values of surface roughness for various terrain types.

Warning errors: SFC_ROUGH_HGT $\geq$ 2.0
Fatal errors: SFC_ROUGH_HGT $\leq$ 0.0
SFC_ROUGH_HGT > 5.0

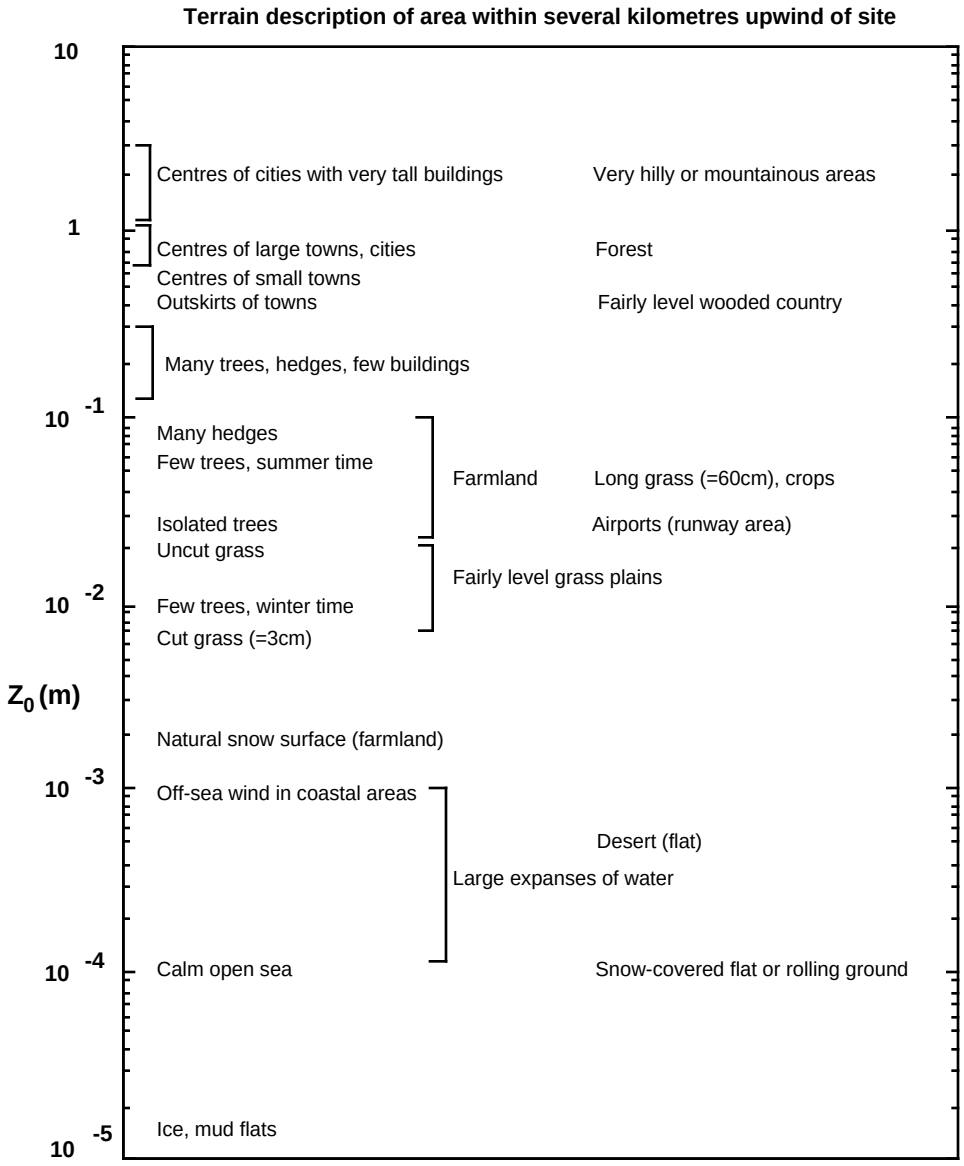


Figure III.B.3. z_0 values for typical terrain types (After ESDU 72026, 1972)

STNLOCPSL

Flag ('Y', 'N') indicating whether or not to use the problem station library. When running MEDIC on the Alpha, STNLOCPSL also controls whether an OBSERV.MET is created from the Problem Metdata Files.

(Default: 'N' -> either no station location data is required or it will be provided in STNLOC.MET)

TERRAIN_ADJUSTMENT (MEDIC/HMEDIC variable)

Flag ('Y', 'N') specifying whether the winds should be terrain shifted ('Y') or zeroed inside the terrain ('N').

(Default : 'Y' for MEDIC and 'N' for HMEDIC)

VECTOR_PLOT_FREQ (MEDIC/HMEDIC variable)

Integer value controlling the density of vectors on the vector plots. A value of one (1) indicates that a vector is plotted at every horizontal grid point. A value of two (2) indicates that a vector is plotted at every other grid point, etc. (Default : VECTOR_PLOT_FREQ = 2)

Warning errors: None
Fatal errors: VECTOR_PLOT_FREQ < = 0

XPROFILE (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of floating point values representing the x-coordinate of the sample profile locations. XPROFILE/YPROFILE constitute a list of the map coordinates of the sample profiles to be plotted. The first value in XPROFILE is matched with the first value in YPROFILE to give the horizontal grid indices of the first profile. Succeeding values from each list are matched in a similar fashion. The XPROFILE/YPROFILE pair is interpolated to the nearest horizontal grid location. The map coordinates in arrays XPROFILE/YPROFILE will in all instances override the values in IPROFILE/JPROFILE, including defaults. If no values are entered for XPROFILE/YPROFILE, the values in IPROFILE/JPROFILE are used.

YPROFILE (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of floating point values representing the y-coordinate of the sample profile locations. See XPROFILE for a full description.

III.B.4b. MEDIC_PARAMS namelist

ALL_WINDS_DIR (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL_ALL) of floating point values representing the wind direction profile for a uniform wind field. Uniform wind fields are created if the direction (ALL_WINDS_DIR), speed (ALL_WINDS_SPD), and optionally height (ALL_WINDS_HGT) are set. No met station information (i.e., problem station library or STNLOC.MET) is required. Interpolation is done linearly, with rotation sense determined by the MEDIC parameter MAX_VEER_PARAM_VERT, and following existing direction interpolation rules. Constant winds relative to the grid may be created by setting the TOPOG_PARAMS namelist parameter MAP_ADJUSTMENT = 'N'. Otherwise constant winds relative to geographic coordinates are assumed (MAP_ADJUSTMENT ='Y'). NOTE #1: In order to achieve circumpolar winds in HMEDIC, you can set ALL_WINDS_SPD to the speed at 60 deg latitude and leave MAP_ADJUSTMENT = 'Y'; ALL_WINDS_DIR has no influence in this case. NOTE #2: When using the ALL_WINDS... parameters, there should not be a METDATASET nor a PROBMET_DST... parameters set.

ALL_WINDS_HGT (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL_ALL) of floating point values representing the heights (m) above terrain of the levels specified in the ALL_WINDS_SPD and ALL_WINDS_DIR arrays. ALL_WINDS_HGT need not be specified if only a single wind speed and direction are to be used. NOTE: When using the ALL_WINDS... parameters, there should not be a METDATASET nor a PROBMET_DST... parameters set.

ALL_WINDS_SPD (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL_ALL) of floating point values representing the wind speed (m/s) profile for a uniform wind field. Uniform wind fields are created if the direction (ALL_WINDS_DIR), speed (ALL_WINDS_SPD), and optionally height (ALL_WINDS_HGT) are set. No met station information (i.e., problem station library or STNLOC.MET) is required. Interpolation is done linearly, with rotation sense determined by the MEDIC parameter MAX_VEER_PARAM_VERT, and following existing direction interpolation rules. Constant winds relative to the grid may be created by setting The TOPOG_PARAMS namelist parameter MAP_ADJUSTMENT = 'N'. Otherwise constant winds relative to geographic coordinates are assumed (MAP_ADJUSTMENT ='Y'). NOTE #1: In order to achieve circumpolar winds in HMEDIC, you can set ALL_WINDS_SPD to the speed at 60 deg latitude and leave MAP_ADJUSTMENT = 'Y'; ALL_WINDS_DIR has no influence in this case. NOTE #2: When using the ALL_WINDS... parameters, there should not be a METDATASET nor a PROBMET_DST... parameters set.

ATMESDATE (MEDIC/HMEDIC variable)

7-character string representing the date (ddmmmyy) of the ATMES gridded metdata to be used by Medic. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid. Note regarding source of "ATMES gridded metdata": GribFM92 formatted data from 25-Apr-86 to 10-May-86 obtained from ECMWF as part of the ATMES study was run through Bob's code to degrib it to this form:

GRIB_OUT_2.yymmddhhmm - geopotential

GRIB_OUT_3.yymmddhhmm - surface winds

GRIB_OUT_5.yymmddhhmm - upper air winds

MEDIC expects these files to be in the directory:

2dua18:[kfoster.modev.datasets.chernobyl.raw]

ATMESTIME (MEDIC/HMEDIC variable)

8-character string representing the time (hh:mm:ss) of the ATMES gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid. Note regarding source of "ATMES gridded metdata": GribFM92 formatted data from 25-Apr-86 to 10-May-86 obtained from ECMWF as part of the ATMES study was run through Bob's code to degrib it to this form:

GRIB_OUT_2.yymmddhhmm - geopotential

GRIB_OUT_3.yymmddhhmm - surface winds

GRIB_OUT_5.yymmddhhmm - upper air winds

MEDIC expects these files to be in the directory:

2dua18:[kfoster.modev.datasets.chernobyl.raw]

AVGTIME (MEDIC/HMEDIC variable)

13-character string representing the time interval (DDD::hh:mm:ss) which is used to specify the averaging time applicable to a given dataset. This value is not used in the calculation, but is used to construct the fully qualified name of the resulting MEDVEL/MATVEL files. It is a required input.

BL_HGT

Floating point value representing the height (m) of the boundary layer above terrain. The height of the boundary layer is the level at which the wind speed and direction become nearly geostrophic, i.e., change little with height because of the negligible effects of surface drag. Upper air sounding (RAWINS or PIBALS) are the best data source for determining BL_HGT. If no soundings are available, an alternative procedure to obtain BL_HGT is to use Figure III.B.1. BL_HGT as used by the model is the height at which the extrapolated upper air profile matches the observed upper air

profile. The wind speed and direction above BL_HGT matches the observed upper air profile if PROFILE_EXTRAP='Y', but remains constant if PROFILE_EXTRAP='N'.

Warning errors: BL_HGT ≤ 50.0
BL_HGT ≥ 3000.0
Fatal errors: BL_HGT ≤ 0.0
BL_HGT ≥ 5000.0

EXTRAP_EXPNT_BL

Floating point value representing the interpolation exponent for the boundary layer. It is used to interpolate winds within the boundary layer, and is valid between the top of the surface layer (SL_HGT) and the top of the boundary layer (BL_HGT). A more complete discussion of this very important parameter is contained in Section II.C.2.b.

Warning errors: EXTRAP_EXPNT_BL > 10.0
Fatal errors: EXTRAP_EXPNT_BL < 0.0
EXTRAP_EXPNT_BL > 50.0

MAX_VEER_PARAM_VERT

Integer value controlling the sense of the vertical direction extrapolation in the boundary layer, and is only used when PROFILE_EXTRAP = 'N' (the parameterized extrapolation). It is the largest angle, measured in degrees, which will be resolved by veering (i.e., the winds rotate in a clockwise direction with height) rather than by backing.

(Default : 240)

Warning errors: NONE
Fatal errors: MAX_VEER_PARAM_VERT < 0
MAX_VEER_PARAM_VERT >360

MAX_VEER_PROF_HORZ

Integer value controlling the sense of how upper air and surface direction discrepancies are resolved at the top of the surface layer, and is only used when PROFILE_EXTRAP='N' (the profile extrapolation).

(Default : 180)

Warning errors: NONE
Fatal errors: MAX_VEER_PROF_HORZ < 0
MAX_VEER_PROF_HORZ > 360

METDATASET

40-character string designating which met data set in the OBSERV.MET file is to be used for this case (i.e., with this MEDIC_PARAMS namelist). The string must match EXACTLY that in the file OBSERV.MET. Suggested string composition is 'yymmddhhq' for most cases. A given dataset may be used more than once, or not at all, in a given MEDIC run: OBSERV.MET is reprocessed for

every MEDIC_PARAMS namelist processed so that any dataset can be located. The METDATASET parameter is required for Alpha versions of MEDIC; for Vax versions, METDATASET will be ignored if any of the PROBMET_DST ... parameters are present.

(Default: METDATASET = 'unset' = OBSERV.MET file not used)

MET_START_DATE (MEDIC/HMEDIC variable)

7-character string indicating the day (ddmmmyy) for which a MEDVEL wind dataset is to be created.

MET_START_TIME (MEDIC/HMEDIC variable)

8-character string representing the time (hh:mm:ss) for which a MEDVEL wind dataset is to be created.

Warning errors:	None
Fatal errors:	MET_START_TIME < 0
	MET_START_TIME > 2400

NHEDATE (MEDIC/HMEDIC variable)

7-character string representing the date (ddmmmyy) of the Northern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

NHEFCST (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the Northern Hemisphere gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the for the hemispheric gridded data are 00, 12, 24, 36, 48, and 60. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

NHETIME (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the Northern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

NODDSDATE (MEDIC/HMEDIC variable)

7-character string representing the date (ddmmmyy) of the NODDS gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

NODDSFCST (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the NODDS gridded metdata to be used by MEDIC. For example, a value of -1 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the for the NODDS gridded data are -1, 12, 24, 36, 48, 60, 72, 84, 96, 108, and 120. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

NODDSTIME (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the NODDS gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

PROBMET_DST

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files (surface, tower, and upper air) if the three Data Set Times are not explicitly given. PROBMET_DST is used as the default for a missing Data Set Time if any, but not all, of PROBMET_DST_SFC, PROBMET_DST_UPR, or PROBMET_DST_TOW are specified. When used simultaneously with the METDATASET parameter, METDATASET is ignored on the VAX and PROBMET_DST is ignored on the Alpha. (However, PROBMET_DST is used by the Alpha MEDIC preprocessor to create an OBSERV.MET file.) This avoids the need for different versions of MEDIC.NML on the VAX and the Alpha.

PROBMET_DST_SFC

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata File for surface data. PROBMET_DST_SFC will override PROBMET_DST for surface datasets. NOTE: When used simultaneously with the METDATASET parameter; METDATASET is ignored on the VAX and PROBMET_DST_SFC is ignored on the Alpha. (However, PROBMET_DST_SFC is used by the Alpha MEDIC preprocessor to create an OBSERV.MET file.) This avoids the need for different versions of MEDIC.NML on the VAX and the Alpha.

PROBMET_DST_TOW

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files for tower data. PROBMET_DST_TOW will override PROBMET_DST for tower datasets. When used simultaneously with the METDATASET parameter; METDATASET is ignored on the VAX and PROBMET_DST_TOW is ignored on the Alphas. (However, PROBMET_DST_TOW is used by the Alpha MEDIC preprocessor to create an OBSERV.MET file.) This avoids the need for different versions of MEDIC.NML on the VAX and the Alphas.

PROBMET_DST_UPR

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files for upper air data. PROBMET_DST_UPR will override PROBMET_DST for upper air datasets.. When used simultaneously with the METDATASET parameter; METDATASET is ignored on the VAX and PROBMET_DST_UPR is ignored on the Alphas. (However, PROBMET_DST_UPR is used by the Alpha MEDIC preprocessor to create an OBSERV.MET file.) This avoids the need for different versions of MEDIC.MNL on the VAX and the Alphas.

PROFILE_EXTRAP

Flag ('Y', 'N') specifying if upper air profiles are used to perform the vertical extrapolation. If PROFILE_EXTRAP is 'N', only the extrapolation parameters are used to generate three-dimensional gridded winds from two-dimensional grids of winds. (Also, see BL_HGT concerning the relationship between PROFILE_EXTRAP and BL_HGT.) If gridded upper air data is used, profile extrapolation is generally preferred.

(Default : 'N', => use parameterized profile method)

PWR_LAW_EXPNT_SL

Floating point value representing the power law exponent for the surface layer. It can be calculated directly from surface-layer wind observations in the vertical using the power law formula given by Eq. (II.C.1). Alternatively, it can be obtained as a function of surface roughness height (surface roughness height is approximately 1/30th of the average height of features such as buildings, grass, trees) as shown in Fig. III.B.1, and the following correction for atmospheric stability applied:

Stable:	add 0.1
Neutral:	as is
Unstable:	subtract 0.1

PWR_LAW_EXPNT_SL is used within the code to extrapolate: (1) wind speeds from their measured height to REF_HGT (see below), (2) wind speeds from the SFC_ROUGH_HGT to the top of the surface layer above terrain in the parameterized extrapolation procedure (see SL_HGT below), and (3) wind speeds from the SFC_ROUGH_HGT and REF_HGT in the profile extrapolation procedure (see Section II.C.2.b for details).

Warning errors:	PWR_LAW_EXPNT_SL > 0.5
Fatal errors:	PWR_LAW_EXPNT_SL < 0.0
	PWR_LAW_EXPNT_SL > 10.0

REF_HGT

Floating point value representing the height (m) above terrain to which the surface wind observations are adjusted using the surface power law (PWR_LAW_EXPNT_SL). Adjustments of the measurements to a common height, usually the typical height of the surface instruments, permits the extrapolation of winds in the horizontal direction.

Warning errors:	REF_HGT ≤ 2.0
	REF_HGT ≥ 100.0
Fatal errors:	REF_HGT ≤ 0.0
	REF_HGT ≥ 500.0

RWMDATE

7-character string specifying the valid date (ddmmmyy) of the Relocatable Window Model gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

RWMFCST

Integer specifying the forecast period (hh) of the Relocatable Window Model gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the Relocatable Window Model are 00, 06, 24, and 36. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

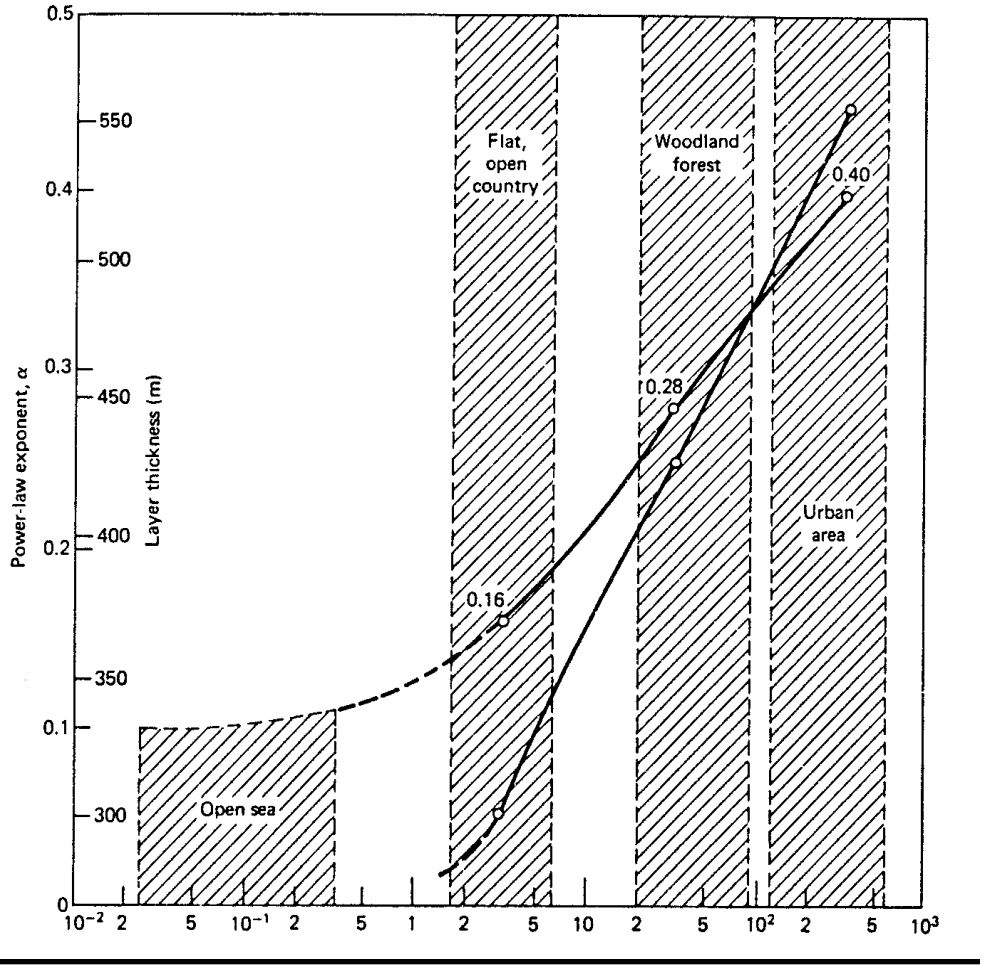


Figure III.B.1 **Modeling the planetary boundary layer (plot from Davenport 1965)**

RWMTIME

8-character string specifying the valid time (hh:mm:ss) of the Relocatable Window Model gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

SHEDATE (MEDIC/HMEDIC variable)

7-character string specifying the valid date (ddmmmyy) of the Southern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

SHEFCST (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the Southern Hemisphere gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of

24 refers to a 24-hour forecast. Current possible values for the hemispheric gridded data are 00, 12, 24, 36, and 48. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

SHETIME (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the Southern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

SINGLE_PROFILE_WINDS

8-character string specifying the upper air metdata station id for which the upper air metdata profile from the Problem Metdata File (met dataset time specified by the PROBMET_DST... parameters) is to be used to construct a uniform wind field. Interpolation is done linearly, with rotation sense determined by the MEDIC parameter MAX_VEER_PARAM_VERT, and following existing direction interpolation rules.

(Default: SINGLE_PROFILE_WINDS = 'unset')

SL_HGT

Floating point value representing the height (m) of the surface layer above terrain. The surface boundary layer is defined as that part of the atmosphere boundary layer where momentum, heat, and energy fluxes are constant. While the height of this layer is typically about 10% of the boundary layer height (BL_HGT), it is more useful to consider SL_HGT as the level at which the wind direction begins to turn with height.

Warning errors:	SL_HGT $\leq$ 2.0
	SL_HGT $\geq$ 100.0
Fatal errors:	SL_HGT $\leq$ 0.0
	SL_HGT $\geq$ 500.0

USE_UPR_AT_SFC

Flag ('Y' or 'N') which controls whether the lowest level of upper air observations from all sources which are also within the surface layer is used to construct the reference level (surface) winds.

(Default : USE_UPR_AT_SFC = 'N')

SAMPLE MEDIC.NML INPUT FILE FOR HEMISPHERIC CASE

```
$medic_control                                  beginning of medic control namelist
medic_vlst          = 'new'          ←  flag controlling creation of MEDIC.VLST file
print_windfields    = 'Y'            ←  flag controlling output of winds
$end                                 ←  end of medic control namelist
$medic_params                                   beginning of medic params namelist
met_start_date      = '01APR85'      ←  date(ddMMMyy) of medvel file to be created
met_start_time      = '0600'         ←  time (hhmmss) of medvel file to be created
avgtime             = '0100'         ←  averaging time for medvel file to be created
nhedate             = '01APR85'      ←  date (ddMMMyy) of AFGWC met dataset to be used
nhefcst             = 00             ←  analysis (00hr) AFGWC data to be used
nhetime             = '1200'         ←  time ('hhmmss') of AFGWC met dataset to be used
$end                                 ←  end of medic params namelist
```

SAMPLE MEDIC INPUT FILE USING OBSERV.MET DATA

```
$medic_control                                  beginning of control namelist
medic_vlst          = 'new'          ←  flag controlling creation of MEDIC.VLST file
print_windfields    = 'Y'            ←  prints wind components
sfc_rough_hgt       = 0.1            ←  surface roughness height
stnlocpsl           = 'N'            ←  required for Alpha versions
$end                                 ←  end of control namelist

$medic_params                                   beginning of params namelist
metdataset          = 'SET1'         ←  pointer id for met dataset in OBSERV.MET
met_start_date      = '01APR85'      ←  date(ddMMMyy) of medvel file to be created
met_start_time      = '0600'         ←  time (hhmmss) of medvel file
avgtime             = '0100'         ←  averaging time for medvel file
sl_hgt              = 25.0           ←  height of the surface layer above the terrain
bl_hgt              = 1400.0         ←  height of the boundary layer above the terrain
pwr_law_expnt_sl    = 0.30           ←  exponent for surface layer power law
extrap_expnt_bl     = 0.30           ←  extrapolation exponent for boundary layer
profile_extrap      = 'N'            ←  method of extrapolation
ref_hgt             = 10.0           ←  hgt above terrain to which sfc winds are adjusted
$end                                 ←  end of params namelist
```

SAMPLE MEDIC.NML INPUT FILE USING PROBLEM MET DATA

```
$medic_control                                  beginning of medic control namelist
medic_vlst          = 'new'          ←  flag controlling creation of MEDIC.VLST file
print_windfields    = 'Y'            ←  prints wind components
sfc_rough_hgt       = 0.1            ←  surface roughness height
stnlocpsl           = 'Y'            ←  required for Alpha versions
$end                                 ←  end of medic control namelist

$medic_params                                   beginning of medic params namelist
metdataset          = '01APR85/06:00'←  required for Alpha versions
probmet_dst         = '850401060'    ←  specification of problem met data file
met_start_date      = '01APR85'      ←  date(ddMMMyy) of medvel file to be created
met_start_time      = '0600'         ←  time (hhmmss) of medvel file to be created
avgtime             = '0100'         ←  averaging time for medvel file to be created
sl_hgt              = 25.0           ←  height of the surface layer above the terrain
bl_hgt              = 1400.0         ←  height of the boundary layer above the terrain
pwr_law_expnt_sl    = 0.30           ←  exponent for surface layer power law
extrap_expnt_bl     = 0.30           ←  extrapolation exponent for boundary layer
```

```
profile_extrap     =  'N'          ←  method of extrapolation
ref_hgt            =  10.0         ←  hgt above terrain to which sfc winds are adjusted
$end                               ←  end of medic params
```

SAMPLE MEDIC INPUT FILE USING ALL_WINDS

```
$medic_control                     beginning of control namelist
medic_vlst         =  'new'        ←  flag controlling creation of MEDIC.VLST file
print_windfields   =  'Y'          ←  prints wind components
sfc_rough_hgt      =  0.1          ←  surface roughness height
stnlocpsl          =  'N'          ←  no problem station library info required
$end                               ←  end of control namelist

$medic_params                      beginning of params namelist
met_start_date     =  '01APR85'    ←  date(ddMMMyy) of medvel file to be created
met_start_time     =  '0600'       ←  time (hhmmss) of medvel file
avgtime            =  '0100'       ←  averaging time for medvel file
all_winds_hgt      =  10. 100. 500.←  array of heights AGL (m)
all_winds_dir      =  160.180. 182 ←  array of wind directions (deg)
all_winds_spd      =  1.5 2.0 2.3  ←  array of wind speeds (m/s)
sl_hgt             =  25.0         ←  height of the surface layer above the terrain
bl_hgt             =  1400.0       ←  height of the boundary layer above the terrain
pwr_law_expnt_sl   =  0.30         ←  exponent for surface layer power law
extrap_expnt_bl    =  0.30         ←  extrapolation exponent for boundary layer
profile_extrap     =  'N'          ←  method of extrapolation
ref_hgt            =  10.0         ←  hgt above terrain to which sfc winds are adjusted
$end                               ←  end of params namelist
```

SAMPLE MEDIC.NML INPUT FILE USING SINGLE_PROFILE_WINDS = 'metstnid'

```
$medic_control                     beginning of medic control namelist
medic_vlst         =  'new',       ←  flag controlling creation of MEDIC.VLST file
print_windfields   =  'Y'          ←  prints wind components
sfc_rough_hgt      =  0.1          ←  surface roughness height
stnlocpsl          =  'Y'          ←  required for Alpha versions
$end                               ←  end of medic control namelist

$medic_params                      beginning of medic params namelist
metdataset         =  '01APR85/06:00'  ←  required for Alpha versions
probmet_dst        =  '850401060'  ←  specification of problem met data file
met_start_date     =  '01APR85'    ←  date(ddMMMyy) of medvel file to be created
met_start_time     =  '0600'       ←  time (hhmmss) of medvel file to be created
avgtime            =  '0100'       ←  averaging time for medvel file to be created
single_profile_winds = 'ARACTES1'  ←  use single wind profile for station ARACTES1 from
                                      specified probmet_dst file
sl_hgt             =  25.0         ←  height of the surface layer above the terrain
bl_hgt             =  1400.0       ←  height of the boundary layer above the terrain
pwr_law_expnt_sl   =  0.30         ←  exponent for surface layer power law
extrap_expnt_bl    =  0.30         ←  extrapolation exponent for boundary layer
profile_extrap     =  'N'          ←  method of extrapolation
ref_hgt            =  10.0         ←  hgt above terrain to which sfc winds are adjusted
$end                               ←  end of medic params
```

III.B.5. STNLOC.MET

The STNLOC.MET file is the input file containing the met station information for the MEDIC code when the Problem Station Library file is not used. It specifies meteorological measurement stations, both surface and upper air, by name, location, and (for the surface stations) measurement height. Each meteorological measurement station referenced in either the Problem Metdata Files or the OBSERV.MET file must have been assigned a location in the STNLOC.MET file if there is no Problem Station Library. While the STNLOC.MET file normally is left unchanged for a well-characterized site, it can be changed at any time between MEDIC runs to provide locations for additional meteorological stations that have become available during an accident response.

This file is a Fortran list-directed input file (rather than a Fortran namelist file). Preceding station location records is a keyword identifying which type of station follows: SFC or UPR, corresponding to surface or upper air meteorological data stations. The records following the keyword have no identifying names, but order is important. The data to specify a surface location is the name (8 characters), x coordinate, y coordinate, and height. An upper air location is similarly specified, except a height is not specified.

The following variables must be assigned a value in the STNLOC.MET file.

Surface Location Record

There is one record per surface station (maximum of 200), and the record consists of four items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. The array for this value in the code is NAMESFC.
2. The Model x coordinate of the surface station location in kilometers. The array for this value in the code is SFCSTX.
3. The Model y coordinate of the surface station location in kilometers. The array for this value in the code is SFCSTY.
4. The height of the instrument above ground level in meters. The array for this value in the code is SFCHGT.

   Fatal errors: SFCHGT $\geq$ 100.0
   SFCSTX $<$ -45000.0
   SFCSTY $<$ -45000.0
   SFCHGT $\leq$ 0.0
   SFCSTX $\geq$ 45000.0
   SFCSTY $\geq$ 45000.0
   SFCHGT $\geq$ 10000.0

Upper Air Location Record

There is one record per upper air station (maximum of 100), and the record consists of three items:

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. The array for this value in the code is NAMEUPR
2. The Model x coordinate of the upper air station location in kilometers. The array for this value in the code is UPRSTX.
3. The Model y coordinate of the upper air station location in kilometers. The array for this value in the code is UPRSTY.

   Fatal errors: UPRSTX < -45000.0
   UPRSTY< -45000.0
   UPRSTX≥ 45000.0
   UPRSTY≥ 45000.0

No entry is required for the number of either surface or upper air records; a count of valid stations of each type is done in the input processing.

SAMPLE STNLOC.MET FILE

UPR				
	'745040'	544.196	4150.254	
	'OAK'	569.007	4178.162	
SFC				
	'HWD'	577.921	4167.145	6.1
	'LVK'	604.317	4172.984	6.1
	'OAK'	569.022	4176.308	6.1
	'NGZ'	560.170	4181.787	6.1
	'SCK'	653.864	4195.958	6.1
	'CCR'	583.429	4204.182	6.1
	'ARACTES1'	613.496	4171.685	10.0
	'LIVELLNL'	614.661	4172.278	10.0

III.B.6. OBSERV.MET

OBSERV.MET is the input file used when meteorological data is not obtained from the Problem Metdata Files. The file is constructed to be read as a Fortran list-directed file, as was STNLOC.MET. An OBSERV.MET file can contain many individual meteorological data sets, called a metdatset. Each such set of data is marked by the occurrence of the keyword METDATASET, followed on the same line by the name (up to 40 characters), in apostrophes, by which it is to be referred in this MEDIC run. Following the METDATASET identifier record is either of the keywords UPR or SFC, which denote that the meteorological data to follow is either upper air or surface data respectively. Following the type-identifier record are individual data records. When the keywords UPR or SFC are again encountered, then input processing toggles to that type. When the METDATASET keyword is encountered, a new set of data is deemed to begin, terminating the previous input one. The format of both types of individual data records is now given.

Surface Met Record

There is one record for each surface observation, and this record consists of three items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. This station identifier name must match exactly with one previously input as a surface station via the STNLOC.MET file or from the Problem Station Library.
2. The wind direction (degrees) is entered. If no wind direction data (i.e. this record is only for a wind speed entry), enter -1.
3. The wind speed (meters/sec) is entered. If no wind speed data (i.e., this record is only for a wind direction entry), enter -1. A zero wind speed is assigned a -1. direction.

Upper Air Met Record

There is one record for each upper air level (maximum of 35 levels per upper air station), and the record consists of four items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. This station identifier name must match exactly with one previously input as an upper air station via the STNLOC.MET file or from the Problem Station Library. This must be specified for the first upper air level, while subsequent levels/records for the same station may simply contain a comma instead of the station identifier. (See example on next page)
2. The observation height, in meters (required).
3. The wind direction (degrees) is entered. If no wind direction data (i.e. this record is only for a wind speed entry), enter -1.
4. The wind speed (meters/sec) is entered. If no wind speed data (i.e., this record is only for a wind direction entry), enter -1. A zero wind speed is assigned a -1. direction.
5. Note: sigma theta data is entered only via the SIGMA_THETA_DATA namelist for ADPIC.

SAMPLE OBSERV.MET FILE

METDATASET	'SET1'				
UPR					
	'OAK'		10.0	270.0	1.0
		,	500.0	270.0	3.0
		,	900.0	270.0	3.0
		,	1400.0	270.0	3.0
SFC					
	'CCR'			270.	2.5
	'HWD'			325.	1.0
	'LVK			270.	1.0
	'SCK'			350.	1.0
	'LIVELLNL'			250.	1.0

III.B.7. MATHEW.NML

MATHEW.NML is the input file that controls a CG-MATHEW run. A basic MATHEW.NML file is a list of the CG-MATHEW stabilities (or SIGMA_VERT values) and the date/times when they take effect. The MATHEW.NML file may also contain a list of the MEDVEL*.BIF files to be processed although these are more typically specified in a MEDIC.VLST file.

The following is a list of the MATHEW.NML variables grouped alphabetically within namelist blocks:

III.B.7.a. MATHEW_CONTROL namelist

HORZ_VECTOR_SCALE

Floating point value which scales the size of the vectors on the vector plots in the x direction.

(Default: HORZ_VECTOR_SCALE = 0.5).

Warning errors: HORZ_VECTOR_SCALE $\geq$ 25.0

Fatal errors: HORZ_VECTOR_SCALE $\leq$ 0.0

IPROFILE

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard R41 grid) of the i indices of the sample profiles of the adjusted winds to be plotted. The first value in IPROFILE is matched with the first value in JPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for IPROFILE with a standard 51 by 51 by 15 MATHEW grid is: 14 26 38 14 26 38 14 26 38. This gives the normal nine sample profiles. Any profiles specified by IPROFILE and JPROFILE (along with the defaults) are superceded by any locations specified by XPROFILE and YPROFILE.

ISLICE

Array (max size = IMAXMAT = 51) of integers (1 to 51 for standard R41 grid) of i-indices at which vector plots of $v - w$ component winds are to be produced. The default value is no slices. Any slices specified by ISLICE are superceded by any locations specified by XSLICE.

ITER_MESSAGE

Integer value specifying the interval number of iterations at which status messages are sent to the terminal. The message gives the number of iterations and the number of "cells" that must still converge.

(Default: ITER_MESSAGE = 50).

Warning errors: None

Fatal errors: ITER_MESSAGE < 0

JPROFILE

Array (max size = MAXPROF = 10) of integers (1 to 51 for a standard R41 grid) of the *j* indices of the sample profiles of the adjusted winds to be plotted. The first value in JPROFILE is matched with the first value in IPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for JPROFILE with a standard 51 by 51 by 15 MATHEW grid is: 14 14 14 26 26 26 38 38 38. This gives the normal nine sample profiles. Any profiles specified by IPROFILE and JPROFILE (along with the defaults) are superceded by any locations specified by XPROFILE and YPROFILE.

JSLICE

Array (max size = JMAXMAT = 51) of integers (1 to 51 for a standard R41 grid) of *j* indices at which vector plots of $u - w$ component winds are to be produced. The default value is no slices. Any slices specified by JSLICE are superceded by any locations specified by YSLICE.

OUTPUT_DESTINATION_DIR

10-character string ('WINDS', 'DISPERSION', 'VLST', or 'GRID') which directs all output from MATHEW (except the MATHEW.LOG file) to a specified directory.

(Default: OUTPUT_DESTINATION_DIR = 'WINDS')

PLOT_LEVELS_ABOVE_TOPO

Array (max size = KMAXMAT = 15) of integers (1 to 15 for a standard R41 grid) representing the *k* levels above terrain at which horizontal vector plots are to be produced.

(Default: PLOT_LEVELS_ABOVE_TOPO = 2 -> produce vector plot at 1 *DELZ above terrain)

PLOT_GRID_LEVELS

Array (max size = KMAXMAT = 15) of integers (1 to 15 for a standard R41 grid) representing the *k* levels above grid bottom (thus some areas may contain vectors of zero length corresponding to grid points underground) at which horizontal vector plots are to be produced. The default is no plots.

PRINT_ADJUSTWIND_ADPIC

Flag ('Y', 'N') specifying whether or not ADPIC winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_ADJUSTWIND_ADPIC = 'N' -> don't print ADPIC winds in MATHEW.LOG)

PRINT_ADJUSTWIND_DIVERG

Flag ('Y', 'N') specifying whether or not the divergence of the adjusted grid point ADPIC winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_ADJUSTWIND_DIVERG = 'N' -> don't print the divergence of the adjusted grid point ADPIC winds in the MATHEW.LOG file)

PRINT_ADJUSTWIND_FACE

Flag ('Y', 'N') specifying whether or not the adjusted face-centered winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_ADJUSTWIND_FACE = 'N' -> don't print these winds in the MATHEW.LOG)

PRINT_ADJUSTWIND_GRID

Flag ('Y', 'N') specifying whether or not the grid point adjusted winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_ADJUSTWIND_GRID = 'N' -> don't print these winds in the MATHEW.LOG)

PRINT_CONVERGENCE

Flag ('Y' or 'N') specifying whether or not the convergence status is to be displayed to the screen during iteration.

(Default: PRINT_CONVERGENCE = 'Y')

PRINT_ICCG_CHECK

Flag ('Y' or 'N') which controls the printing of the divergence checks on the conjugate-gradient solution.

(Default = PRINT_ICCG_CHECK = 'N')

PRINT_INITIALWIND_DIVERG

Flag ('Y', 'N') specifying whether or not the divergence of the extrapolated winds is to be printed in the MATHEW.LOG file.

(Default: PRINT_INITIALWIND_DIVERG = 'N' -> don't print the divergence in the MATHEW.LOG)

PRINT_INITIALWIND_FACE

Flag ('Y', 'N') specifying whether or not the face-centered extrapolated winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_INITIALWIND_FACE = 'N' -> don't print these winds in the MATHEW.LOG)

PRINT_INITIALWIND_GRID

Flag ('Y', 'N') specifying whether or not the grid point extrapolated winds are to be printed in the MATHEW.LOG file.

(Default: PRINT_INITIALWIND_GRID = 'N' -> don't print these winds in the MATHEW.LOG)

PRINT_LAMBDAS

Flag ('Y', 'N') specifying whether or not the Lagrangian multipliers are to be printed in the MATHEW.LOG file.

(Default: PRINT_LAMBDAS = 'N' -> don't print the multipliers in the MATHEW.LOG)

PRINT_SIDES

Flag ('Y' or 'N') which controls the printing of the three-dimensional arrays identifying the cell side classifications.

(Default: PRINT_SIDES = 'N')

VECTOR_PLOT_FREQ

Integer value controlling the density of vectors on the vector plots. A value of one (1) indicates that a vector is plotted at every horizontal grid point. A value of two indicates that a vector is to be plotted every other grid point, etc.

(Default: VECTOR_PLOT_FREQ = 2)

Warning errors: None

Fatal errors: VECTOR_PLOT_FREQ < 0

VERT_VECTOR_SCALE

Floating point value which scales the size of the vectors on the vector plots in the z direction.

(Default: VERT_VECTOR_SCALE = 25.0)

Warning errors: VERT_VECTOR_SCALE $\geq$ 200.0

Fatal errors: VERT_VECTOR_SCALE $\leq$ 0.0

VLST_OPTION

6-character string ('NONE', 'NEW', 'MODIFY') specifying how MATHEW is to handle output of a VLST file. If it is set to 'NONE ' then MATHEW does not produce a MATHEW.VLST file, this is the default. If it is 'NEW ' then MATHEW produces a MATHEW.VLST file that supercedes any preceeding versions. If it is set to 'MODIFY' then MATHEW reads the MATHEW.VLST file currently available (if it exists) and the overwrites MATVEL filenames that match those being processed in the current run except for the sequence number and appends new filenames to the end of the MATHEW.VLST records.

XPROFILE

Array (max size = MAXPROF = 10) of integers (1 to 51 for a standard R41 grid) of the x coordinates of the sample profiles of the adjusted winds to be plotted. The first value in XPROFILE is matched with the first value in YPROFILE to give a horizontal location for the first profile. The

profile location plotted is the *i,j* grid location nearest to the specified *x,y* coordinates. Succeeding coordinates from each list are matched in a similar fashion. Values for XPROFILE, YPROFILE supercede any user settings for IPROFILE and JPROFILE as well as the nine default profiles (see IPROFILE and JPROFILE for the defaults). Each XPROFILE value must lie within the MATHEW grid.

XSLICE

Array (max size = IMAXMAT = 51 for a standard R41 grid) of floating point values representing the *x* coordinates of the vertical planes of $v - w$ winds vectors of the adjusted winds to be plotted. The slice location plotted is the *i* grid location nearest to the specified *x*coordinate. Values for XSLICE supercede any user settings for ISLICE. The default is no slice plots. Each XSLICE value must lie within the MATHEW grid.

YPROFILE

Array (max size = MAXPROF = 10) of integers (1 to 51 for a standard R41 grid) of the *y* coordinates of the sample profiles of the adjusted winds to be plotted. The first value in YPROFILE is matched with the first value in XPROFILE to give a horizontal location for the first profile. The profile location plotted is the *i,j* grid location nearest to the specified *x,y* coordinates. Succeeding coordinates from each list are matched in a similar fashion. Values for XPROFILE, YPROFILE supercede any user settings for IPROFILE and JPROFILE as well as the nine default profiles (see IPROFILE and JPROFILE for the defaults). Each YPROFILE value must lie within the MATHEW grid.

YSLICE

Array (max size = JMAXMAT = 51 for a standard R41 grid) of floating point values representing the *y* coordinates of the vertical planes of $u - w$ winds vectors of the adjusted winds to be plotted. The slice location plotted is the *j*grid location nearest to the specified *y*coordinate. Values for YSLICE supercede any user settings for JSLICE. The default is no slice plots. Each YSLICE value must lie within the MATHEW grid.

III.B.7.b. MATHEW_PARAMS namelist

CONV_TOLERANCE

Floating point value specifying the required convergence tolerance limit for the conjugate-gradient solver.

(Default: CONV_TOLERANCE = 0.01)

ELLIPTIC_EQN_TEST

Flag ('Y' or 'N') which controls whether the code sets up and runs an elliptic equation test based on sinusoidal fields to test the implementation of boundary conditions and the conjugate-gradient solver.

(Default: ELLIPTIC_EQN_TEST = 'N')

FIX_LATERAL_INFLOW

Flag ('Y' or 'N') which controls whether the inflow is fixed on all lateral faces.

(Default: FIX_LATERAL_INFLOW = 'N')

FIX_XMAX_INFLOW

Flag ('Y' or 'N') which controls whether the inflow is fixed on the maximum index x face. Overridden if FIX_LATERAL_INFLOW is set.

(Default: FIX_XMAX_INFLOW = 'N')

FIX_XMIN_INFLOW

Flag ('Y' or 'N') which controls whether the inflow is fixed on the minimum index x face. Overridden if FIX_LATERAL_INFLOW is set.

(Default: FIX_XMIN_INFLOW = 'N')

FIX_YMAX_INFLOW

Flag ('Y' or 'N') which controls whether the inflow is fixed on the maximum index y face. Overridden if FIX_LATERAL_INFLOW is set.

(Default: FIX_YMAX_INFLOW = 'N')

FIX_YMIN_INFLOW

Flag ('Y' or 'N') which controls whether the inflow is fixed on the minimum index y face. Overridden if FIX_LATERAL_INFLOW is set.

(Default: FIX_YMIN_INFLOW = 'N')

ITER_HALT

Integer value specifying the maximum allowed number of iterations for the conjugate gradient solver.

(Default: ITER_HALT = 1500)

LATERAL_BOUNDARY

15-character string('ADJUST_NORMAL', 'FIXED_NORMAL', or 'MIXED') specifying the boundary condition on all lateral faces.

(Default: LATERAL_BOUNDARY = 'ADJUST_NORMAL')

MEDVEL_NAMES

Array (max size = MAX_VEL = 10000) of 40-character strings that specify the MEDVEL***.BIF files to be processed by the current MATHEW run. Each string is a fully qualified, and thus unique, filename. The list of files entered via MEDVEL_NAMES supercedes the list of files specified in the MEDIC.VLST file, an alternative way of indicating those MEDVEL***.BIF files to be processed.

MIXED_LATERAL_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on all lateral faces.

(Default: MIXED_LATERAL_ADJUST = 1.0)

MIXED_TOP_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the top face.

(Default: MIXED_TOP_ADJUST = 1.0)

MIXED_XMAX_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index x face. Overridden if MIXED_LATERAL_ADJUST is set.

(Default: MIXED_XMAX_ADJUST = 1.0)

MIXED_XMIN_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index x face. Overridden if MIXED_LATERAL_ADJUST is set.

(Default: MIXED_XMIN_ADJUST = 1.0)

MIXED_YMAX_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index y face. Overridden if MIXED_LATERAL_ADJUST is set.

(Default: MIXED_YMAX_ADJUST = 1.0)

MIXED_YMIN_ADJUST

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index y face. Overridden if MIXED_LATERAL_ADJUST is set.

(Default: MIXED_YMIN_ADJUST = 1.0)

PARAMS_DATE

Array (max size = MAX_TIMES = 10000) of 7-character strings specifying the dates (ddmmmyy) for the time-varying inputs, STAB_MATHEW and SIGMA_VERT, i.e., it determines the date component of the date/time when one or the other of these variables (but not both) change values.

PARAMS_TIME

Array (max size = MAX_TIMES = 10000) of 8-character strings specifying the times (hh:mm:ss) for the time-varying inputs, STAB_MATHEW and SIGMA_VERT, i.e., it determines the time component of the date/time when one or the other of these variables (but not both) change values.

SIGMA_HORZ

Floating point value that, together with SIGMA_VERT, controls how much mass-adjustment takes place in the horizontal as opposed to the vertical direction. Since the ratio of SIGMA_VERT to SIGMA_HORZ controls MATHEW's behavior, SIGMA_HORZ is normally set to 1.0 and only SIGMA_VERT is changed. The default value is 1.0.

Warning errors:	SIGMA_HORZ > 5.0
Fatal errors:	SIGMA_HORZ $\leq$ 0.0
	SIGMA_HORZ $\geq$ 100.0

SIGMA_VERT

Array(max size = MAX_TIMES=10,000) of floating point values that, together with SIGMA_HORZ, control how much mass-adjustment takes place in the vertical as opposed to the horizontal direction. The choice of SIGMA_VERT is a function of the grid cell geometry and the atmospheric stability and can be associated, all be it crudely, with Pasquill categories (see MATHEW input variable STAB_MATHEW below). The assessor may use SIGMA_VERT to gain finer control over the MATHEW adjustment process than is provided via STAB_MATHEW. SIGMA_VERT is a time-varying input like STAB_MATHEW, thus any change in its value must be associated with corresponding entries in the PARAMS_DATE and PARAMS_TIME arrays.

Warning errors: SIGMA_VERT > 2.0

Fatal errors: SIGMA_VERT ≤ 0.0

SIGMA_VERT > 100.0

STAB_MATHEW

Array (max size = MAX_TIMES = 10000) of integers (1-6) representing the Pasquill-Gifford stability classes that, when associated with the PARAMS_DATE and PARAMS_TIME arrays, allows MATHEW to determine the appropriate stability for any given data set it is processing. Thus, STAB_MATHEW can be considered a time-varying input analogous to those in the ADPIC.NML file. These times need only be specified when STAB_MATHEW changes, not necessarily for every data set. STAB_MATHEW is used to control the relative degree of adjustment by assigning values to SIGMA_VERT according to the table below. Thus, SIGMA_VERT is a more direct way of controlling MATHEW's behavior and is also a time-varying input array. The use of SIGMA_VERT directly supercedes the use of STAB_MATHEW.

(Default: STAB_MATHEW = 4 -> neutral stability)

STAB_CLASS	SIGMA_VERT
1	3.3×(2×DELZ/DELX)
2	2.4×(2×DELZ/DELX)
3	1.6×(2×DELZ/DELX)
4	1.0×(2×DELZ/DELX)
5	0.7×(2×DELZ/DELX)
6	0.33×(2×DELZ/DELX)

Fatal errors: STAB_MATHEW < 1

STAB_MATHEW > 6

A sample MATHEW namelist file is shown below:

```
$MATHEW_PARAMS
print_adjustwind_diverg                    =  'Y'      ←  print adjusted divergence
print_initialwind_diverg                   =  'Y'         print initial divergence
vector_plot_freq                           =  1           plot all wind vectors
$end

$MATHEW_PARAMS
MEDVEL_NAMES                               =           ←  datasets to be processed.
'MEDVEL_01APR85_0000_00_000_0100__012.BIF'
'MEDVEL_01APR85_0100_00_000_0100__012.BIF'
'MEDVEL_01APR85_0200_00_000_0100__012.BIF'
'MEDVEL_01APR85_0300_00_000_0100__012.BIF'
'MEDVEL_01APR85_0400_00_000_0100__012.BIF'
'MEDVEL_01APR85_0500_00_000_0100__012.BIF'
'MEDVEL_01APR85_0600_00_000_0100__012.BIF'
'MEDVEL_01APR85_0700_00_000_0100__012.BIF'
'MEDVEL_01APR85_0800_00_000_0100__012.BIF'

PARAMS_DATE    = '01APR85'  '01APR85'      ←  date of new stabilities
PARAMS_TIME    =    '0000'     '0500'      ←  time of new stabilities
STAB_MATHEW    =         3          5      ←  new stabilities
$end
```

TOP_BOUNDARY

15-character string (‘ADJUST_NORMAL’, ‘FIXED_NORMAL’, or ‘MIXED’) specifying the boundary condition on the top face (Default: TOP_BOUNDARY = ‘ADJUST_NORMAL’).

XMAX_BOUNDARY

15-character string ('ADJUST_NORMAL', 'FIXED_NORMAL', or 'MIXED') specifying the boundary condition on the maximum index x face. Overridden if LATERAL_BOUNDARY is set.

(Default: XMAX_BOUNDARY = 'ADJUST_NORMAL')

XMIN_BOUNDARY

15-character string ('ADJUST_NORMAL', 'FIXED_NORMAL', or 'MIXED') specifying the boundary condition on the minimum index x face. Overridden if LATERAL_BOUNDARY is set.

(Default: XMIN_BOUNDARY = 'ADJUST_NORMAL')

YMAX_BOUNDARY

15-character string ('ADJUST_NORMAL', 'FIXED_NORMAL', or 'MIXED') specifying the boundary condition on the maximum index y face. Overridden if LATERAL_BOUNDARY is set.

(Default: YMAX_BOUNDARY = 'ADJUST_NORMAL')

YMIN_BOUNDARY

15-character string ('ADJUST_NORMAL', 'FIXED_NORMAL', or 'MIXED') specifying the boundary condition on the minimum index y face. Overridden if LATERAL_BOUNDARY is set.

(Default: YMIN_BOUNDARY = 'ADJUST_NORMAL')

III.B.8. MEDIC.VLST

MATHEW can optionally read a list of fully qualified, and therefore unique, MEDVEL***.BIF filenames from MEDIC.VLST. This file is produced by MEDIC and contains the list of MEDVEL***.BIF files created by a given MEDIC run. This is an ASCII file that can be edited by an assessor and provides a mechanism to handle multiple MEDVEL groups in a single directory by allowing the user to maintain a series of MEDIC.VLST files (under different names, with the desired file being copied to MEDIC.VLST as needed).

III.B.9. ADPIC.NML

ADPIC.NML is an input file to the ADPIC code which contains the source and editing parameter data for an ADPIC problem. Before ADPIC is run, either initially or as a restart, an ADPIC.NML file must be prepared.

Descriptions of all of the individual ADPIC.NML parameters follows, by the namelists in which they are to be entered:

III.B.9.a ADPIC_CONTROL namelist

DIFF_METHOD

5-character string ('RDM' or 'GRADI') specifying the type of diffusion calculation method to be used.

DIFF_METHOD = 'RDM' refers to the new Random Displacement Method (RDM). For this method, the types of horizontal and vertical turbulence parameterizations to be used for K_Z and K_H respectively, by the RDM must be specified using the ADPIC_METPARAMS namelist parameters TURB_PARAM_VERT and TURB_PARAM_HORZ.

DIFF_METHOD = 'GRADI' refers to ADPIC's original particle-in-cell gradient diffusion method which uses a hybrid Eulerian-Langragian method. The ADPIC_METPARAMS namelist parameter TURB_PARAM_TYPE controls the diffusion coefficients to be used for this diffusion method.

INITIAL_TIME_STEP

Floating point value specifying the value for the initial time step (seconds) for ADPIC; all succeeding time steps will be calculated by the code. ADPIC calculates the succeeding time steps such that a particle should not move more than one cell in that time step. INITIAL_TIME_STEP is also used to generate a warning error ("Time step has decreased to x.xxxExx") whenever code-calculated time steps are less than INITIAL_TIME_STEP/10.

NESTS

Integer (0-4) controlling the number of sampling nests to be used. fewer nests require less buffer zone between the source and the edge of the grid. The default center of the nests will be the least squares center of all the source locations. This center can be overridden with the ADPIC_CONTROL namelist parameters X_NEST_LOC and Y_NEST_LOC.

(Default: NESTS = 4)

PARTPOS

A 1-character string ('N', 'P', 'V') indicating in which format to generate PARTPOS files at the time interval specified by PLOT_INTERVAL:

'N' = no PARTPOS file is to be generated

'P' = PERSPEC format (3 particle position arrays)

'V' = video format (3 particle position arrays, plus source and identifier arrays)

(Default: PARTPOS = 'N' → no PARTPOS files are to be generated)

PK_INTERVAL

13-character string (ddd::hh:mm:ss) specifying the desired concentration averaging interval used in collecting the data for SAMPLING_TYPE = 'PKAIR'. If PK_INTERVAL is not an even multiple of SAMPLING_INTERVAL, then PK_INTERVAL is internally rounded down, if possible, so that it can be exactly divided into SAMPLING_INTERVAL.

(Default: PK_INTERVAL = '000::00:15:00')

Minimum value = 1 minute (smaller values will be changed to 1 minute)

Maximum value = SAMPLING_INTERVAL (larger values will be changed to SAMPLING_INTERVAL by ADPIC)

PLOT_INTERVAL

A 13-character time interval string (ddd::hhmmss) specifying the time interval at which particle plots are written to the graphics file.

(Default: PLOT_INTERVAL = '000::01:00:00' → 1-hour plot interval)

PRINT_ADVECT_VEL

Flag ('Y' or 'N') which controls the printing of advection velocities in echo file as they are read from the MATVEL file.

(Default: PRINT_ADVECT_VEL = 'N' → don't print advection velocities)

PRINT_CELL_CONC

Flag ('Y' or 'N') which controls the printing of cell concentrations in echo file at each SAMPLING_INTERVAL interval.

(Default: PRINT_CELL_CONC = 'N' → print cell concentrations)

PRINT_FRCTN_VEL

Flag ('Y' or 'N') which controls the printing of friction velocities in the echo file at each STATUS_INTERVAL interval.

(Default: PRINT_FRCTN_VEL = 'N' → don't print friction velocities)

PRINT_PART_COORD

Flag ('Y' or 'N') which controls the printing of particle coordinates in echo file at each SAMPLING_INTERVAL interval.

(Default: PRINT_PART_COORD = 'N' → don't print particle coordinates)

PRINT_SAMPLING_BINS

Flag ('Y' or 'N') which controls the printing of sampling bin concentrations to the echo file at each SAMPLING_INTERVAL interval.

(Default: PRINT_SAMPLING_BINS = 'N' → don't print sampling bin concentrations)

PRINT_TURB_PARAM_GRADI

Flag ('Y' or 'N') which controls the printing (to the ADPIC log file) of horizontal and vertical diffusion coefficients (K_H and K_z) at each grid point over the portion of the grid that contains particles.

(Default: PRINT_TURB_PARAM_GRADI = 'N' → don't print Kh and Kz)

PRINT_PART_VEL

Flag ('Y' or 'N') which controls the printing of all velocities in echo file at each STATUS_INTERVAL interval.

(Default: PRINT_PART_VEL = 'N' → don't print velocities)

PRINT_TOPOG

Flag ('Y' or 'N') which controls the printing of topography in echo file at beginning of problem.

(Default: PRINT_TOPOG = 'Y' → print topography)

PROBLEM_TITLE

70-character string specifying the label for the particle plot and other ADPIC output.

REFLECT_AT_GRIDTOP

A value of REFLECT_AT_GRIDTOP = 'Y' (yes) causes particles approaching the computational grid top (due to advection or diffusion) to be reflected back into grid. When the grid top height does not equal the mixed layer height, this is an artificial restriction and is provided as a "safety net"; that is, reflection at the grid top is used to prevent particles from diffusing upward through the top of the grid, being lost to the calculation, and, therefore, unable to diffuse back downward. However, reflection at grid top should not be relied upon to contain these particles. Rather, computational grids should constructed with sufficient depth that they will contain particles of interest in the calculation. For example, for boundary layer problems it is recommended that the grid depth be specified so that the difference between the maximum terrain elevation and the elevation of the top of the grid is at least two times the mixed layer height. Note that this parameter is used *only* with DIFF_METHOD = 'RDM'.

(Default: 'Y').

REFLECT_AT_MLHGT

A value of REFLECT_AT_MLHGT = 'Y' (yes) causes particles approaching a height equal to the grid bottom height plus the mixing layer depth, to be reflected (this applies to both upward- and downward-moving particles). Note that this is not necessarily the mixed layer height above terrain, if terrain elevations vary within the computational grid. This option is recommended for *flat terrain only*, because the wind fields adjusted for terrain effects by MATHEW are not adjusted for this assumed rigid elevated surface, and because ADPIC cannot handle the reflection of particles at mixed layer surfaces if it intersects terrain. This reflection method may be physically realistic when a strong elevated inversion limits vertical mixing at a fixed height above sea level, but may not be realistic for many cases, especially mesoscale or long-range cases or for cases of significant variation in terrain elevation. Also, refer to parameter REFLECT_AT_GRIDTOP. Note: This parameter used *only* with DIFF_METHOD = 'RDM'.

(Default: 'N').

RESTART_DATE

7-character date string (ddMMMyy) the restart date of the ADPIC calculation. This restart date/time MUST have a corresponding ADPIC*.RST and CONC*.RST files (see RESTART_WRITE_INTERVAL description). Also see RESTART_TIME, which specifies the time.

(RESTART_DATE = 'unset' is the default, i.e. ,for a non-restart run)

RESTART_TIME

8-character string specifying the restart time (hh:mm:ss) of the ADPIC calculation. This restart date/time MUST have a corresponding dump in the ADPIC*.RST and CONC*.RST files (see RESTART_WRITE_INTERVAL description). Also see RESTART_DATE, which specifies the date.

(RESTART_TIME = 'unset' is the default, i.e., for a non-restart run)

RESTART_WRITE_DATE

Array (max size = MAX_RESTART = 10) of 7-character strings specifying the dates (ddmmmyy) in increasing chronological order at which data is output from ADPIC into restart dump files (ADPIC*.RST and CONC*.RST). It therefore determines the times at which a problem can be restarted since the dump files contain all of the necessary information to restart the problem. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying too many restart files. RESTART_WRITE_DATE and RESTART_WRITE_TIME override RESTART_WRITE_INTERVAL.

RESTART_WRITE_INTERVAL

13-character string specifying the time interval (ddd::hh:mm:ss) at which data is output from ADPIC into a restart dump files (ADPIC*.RST and CONC*.RST). It therefore determines the times at which a problem can later be restarted since the restart file contains all fo the necessary information, i.e., coordinates of all of the particles. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying an interval which produces a large number of such files. To specify date/times for restart file writes which are not multiples of RESTART_WRITE_INTERVAL, use RESTART_WRITE_DATE and RESTART_WRITE_TIME; RESTART_WRITE_DATE and RESTART_WRITE_TIME override RESTART_WRITE_INTERVAL

No restart file (RESTART_WRITE_INTERVAL = '000::0000') is usually specified for an Initial calculation.

RESTART_WRITE_TIME

Array (max size = MAX_RESTART = 10) of 8-character strings specifying the times (hh:mm:ss) in increasing chronological order at which data is output from ADPIC into restart dump files (ADPIC*.RST and CONC*.RST). It therefore determines the times at which a problem can be restarted since the dump files contain all of the necessary information to restart the problem. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying too many restart files. RESTART_WRITE_DATE and RESTART_WRITE_TIME override RESTART_WRITE_INTERVAL.

RUN_DURATION

13-character string specifying the time interval (ddd::hh:mm:ss) for the current ADPIC run. It is relative to RUN_START_DATE \ RUN_START_TIME if it is not a restart run, and from RESTART_DATE \ RESTART_TIME if it is a restart run. Run_stop_date and run_stop_time be used to specify the length of an ADPIC run in terms of beginning and ending rather than beginning and the duration interval.

RUN_START_DATE

7-character string specifying the start date (ddmmmyy) of the ADPIC calculation. Note that this value must remain unchanged from the original run to a restart run.

RUN_START_TIME

8-character string specifying the start time (hh:mm:ss) of the ADPIC calculation. Note that this value remains unchanged from the original run to a restart run.

RUN_STOP_DATE

7-character string specifying the stop date (ddmmmyy) of the ADPIC calculation. Specifying both RUN_STOP_DATE and RUN_STOP_TIME determines the total duration of an ADPIC execution. RUN_STOP_DATE\RUN_STOP_TIME will override RUN_DURATION.

RUN_STOP_TIME

8-character string specifying the stop time (hh:mm:ss) of the ADPIC calculation. Specifying both RUN_STOP_DATE and RUN_STOP_TIME determines the total duration of an ADPIC execution. RUN_STOP_DATE/RUN_STOP_TIME will override RUN_DURATION.

SAMP_CELL_DELZ_FRACT

Floating point value representing the vertical height of the sampling cell expressed as a fraction of DELZ. This allows finer resolution of concentration in the vertical, provided there is a sufficient number of particles in the sampling volume.

(Default: SAMP_CELL_DELZ_FRACT = 1.0)

SAMPLING_HGT

Array (max size = MAXSBINS = 30) of floating point values specifying the heights (m) above ground at which the output bin contents (determined by SOURCES_TO_SAMPLING_BIN and SAMPLING_TYPE) are to be calculated. For example:

a) if inhalation doses or concentrations are desired, a SAMPLING_HGT value of 1.5m (breathing height) would be used;
b) if concentrations at stack heights or flight levels are desired, a SAMPLING_HGT equal to the height of interest would be used;

c) for ground deposition, a SAMPLING_HGT of 0.0m would be used.

SAMPLING_HGT_REF

11-Character string ('SEA_LEVEL', 'GRID_BOTTOM', 'TERRAIN') specifying which reference level/type would be used for sampling bins.

(Default: SAMPLING_HGT_REF = 'TERRAIN' = sampling hgts are meters above terrain)

SAMPLING_INTERVAL

13-character string specifying the output interval (ddd::hh:mm:ss) at which particle information is written to the CONC file.

(Default: SAMPLING_INTERVAL = '000::0100' → 1-hour dumps to the CONC file)

SAMPLING_TYPE

Array (max size = MAXSBINS = 30) of 6-character strings ('INTAIR', 'INSAIR', 'TOTDEP', AVGAIR', 'PKAIR') representing the type of concentration output desired for the ADPIC output bins:

SAMPLING_TYPE	=	'INTAIR'	→	integrated air concentration
SAMPLING_TYPE	=	'INSAIR'	→	instantaneous air concentration
SAMPLING_TYPE	=	'INTDEP'	→	integrated deposition
SAMPLING_TYPE	=	'TOTDEP'	→	total ground deposition
SAMPLING_TYPE	=	'AVGAIR'	→	SAMPLING_INTERVAL-time-averaged air concentration
SAMPLING_TYPE	=	'PKAIR'	→	composite peak (over SAMPLING_INTERVAL)

Option 'INTDEP' may only be used with the Hybrid Particle ADPIC. Particles reaching the ground are saved in a list that includes their age, deposition time, and deposition position. At sampling times they are assigned their appropriate activity and toxicity and distributed in sampling bins defined on a Eulerian grid. This permits accurate decay in sampling bins that contain multiple course and enables the total time integrated deposition dose to be calculated. Rainout is not included.

Option 'AVGAIR' produces ADPIC bins of SAMPLING_INTERVAL-time-averaged air concentrations. The time averaging is achieved by taking INTAIR bins and dividing by the SAMPLING_INTERVAL.

Option 'PKAIR' computes PK_INTERVAL-time-averaged air concentrations, and stores the peak values encountered over the SAMPLING_INTERVAL time interval; i.e., at a given SAMPLING_INTERVAL time, the output CONC*.BIF file for a bin with SAMPLING_TYPE =

'PKAIR' contains, for a given location, the peak value of PK_INTERVAL-averaged air concentration encountered over successive PK_INTERVAL time intervals from the beginning of the run. The internal arrays collecting 'PKAIR' concentration data are continually updated so that output for successive SAMPLING_INTERVALs will contain the highest short-term concentration to that point in the run. Thus, at the end of a run the concentrations collected under the 'PKAIR' option represents the highest short-term concentration at all locations during the entire simulation.

NOTE: ADPIC diffusivities represent averaging periods of 10 to 60 minutes, with a mean of about 30 minutes. Therefore, if an averaging period (controlled by PK_INTERVAL for PKAIR bins, and SAMPLING_INTERVAL for AVGAIR bins) less than 30 minutes is desired, a power-law scaling is applied internally to convert to the desired shorter averaging period.

For toxic chemical responses, an AVGAIR bin with SAMPLING_INTERVAL equivalent to 1 hr is typically used. Additionally, a PKAIR bin (with PK_INTERVAL set for 15 min and SAMPLING_INTERVAL set for 1 hr) is used to produce plots for concentration-dependent chemicals (i.e., those chemicals which have very short-term health effects, typically 15 minutes or less).

NOTE: PKAIR will use 2 sampling bins — one in its original positioning in the SAMPLING_TYPE array specification, and one at the end of the SAMPLING_TYPE array, the latter being used as a temporary storage for accumulating the peak concentrations. Therefore, if you wish to have 3 PKAIR sampling bins, there will be only 24 (30-3*2) other possible sampling bins available.

SKIP_RANDOM_COUNT

Integer number of random numbers to skip in random number sequence before the first random number is used. This parameter is primarily for programmer's use as a debugging tool: use of SKIP_RANDOM_COUNT can identify how much variability in results is due to randomness.

SOURCES_TO_SAMPLING_BIN

2-D array (max ADPIC sources, max sampling bins) of integers (1 to max ADPIC sources) specifying which ADPIC sources are to be associated with which sampling bin. SOURCES_TO_SAMPLING_BIN(1,2) = 3 indicates that sampling bin 2 includes particles from source 3 (it may also include others, this entry just specifies that part of the sampling strategy). The source indices must be used in sequence for a given sampling bin index. See sample ADPIC.NML files in this document for further examples.

(Default: all elements of SOURCES_TO_SAMPLING_BIN default to 0)

STATUS_INTERVAL

13-character string specifying the edit interval (ddd::hh:mm:ss) at which particle status information lines are written to the screen and to the echo file.

(Default: STATUS_INTERVAL = '000::0100' → 1-hour edit interval)

USE_EXPL_CLD_RISE

Flag ('Y' or 'N') controlling whether or not the explosive cloud rise option is to be used.

Note: the rest of the parameters for the explosive cloud rise are contained in the namelist ADPIC_EXPL_CLD

(Default: USE_EXPL_CLD_RISE = 'N' ⇒ do not use explosive cloud rise)

USE_MOVING_RECEPTORS

Flag ('Y' or 'N') to for the moving receptor capability, which allows sampling to be done at the location of multiple moving "receptors".

(Default: USE_MOVING_RECEPTORS = 'N' = don't use moving receptors)

USE_VLST

Flag ('Y' or 'N') controlling whether the file 'MATHEW.VLST' or 'MEDIC.VLST is to be used to obtain a list of valid MATVEL/MEDVEL files for this run. 'Y' means use the appropriate vlst file, and 'N' means search for '

'MATVEL*.BIF'' or 'MEDVEL*.BIF' in the current MODEL$A_WINDS directory.

(Default: USE_VLST = 'N' -> search for 'MATVEL*.BIF' or 'MEDVEL*.BIF' in the current MODEL$A_WINDS directory)

WIND_TYPE

9-character string ('MATHEW', 'MEDIC', or 'POWER_LAW') controlling whether MATVEL files, MEDVEL files, or analytic power law wind profile files are to be used for input wind velocity data. 'MATHEW' means use MATVEL files, and 'MEDIC' means use MEDVEL files (which have not been adjusted by the MATHEW model. 'POWER_LAW' means use vertical power law wind field in ADPIC which is purely analytical and not gridded (no gridded input wind files are required). Power law winds are controlled by the ADPIC_METPARAMS namelist parameters U_MEAN_REF, U_MEAN_REF_HGT, PWRLAW_EXPNT_U, and U_MEAN_DIR, that must be entered for this option. Winds vary with power law in the vertical, but do not vary in the horizontal directions at any given height. WARNING: Use 'POWER_LAW' only with with flat terrain.

(Default: WIND_TYPE= 'MATHEW' ⇒ use MATVEL files)

X_NEST_LOC

Floating point value representing the x-coordinate about which the nested grid structure will be built. This is especially useful if there are multiple source locations and you wish to override the default center (which is the least squares distance from all of the source locations) of the nested grid structure.

(Default: X_NEST_LOC = -99999.0 = use default center of all source locations.)

X_WINDOW_MAX

Floating point value representing a map coordinate (km) indicating the right-hand X edge of a window to be specified and fixed for outputing dot plots.

(Default: X_WINDOW_MAX = 0 -> use standard windows)

X_WINDOW_MIN

Floating point value representing a map coordinate (km) indicating the left-hand X edge of a window to be specified and fixed for outputing dot plots.

(Default: X_WINDOW_MIN = 0 -> use standard windows)

Y_NEST_LOC

Floating point value representing the y-coordinate about which the nested grid structure will be built. This is especially useful if there are multiple source locations and you wish to override the default center (which is the least squares distance from all of the source locations) of the nested grid stucture.

(Default: Y_NEST_LOC = -99999.0 = use default center of all source locations.)

Y_WINDOW_MAX

Floating point value representing a map coordinate (km) indicating the right-hand Y edge of a window to be specified and fixed for outputing dot plots.

(Default: Y_WINDOW_MAX = 0 -> use standard windows)

Y_WINDOW_MIN

Floating point value representing a map coordinate (km) indicating the left-hand Y edge of a window to be specified and fixed for outputing dot plots.

(Default: Y_WINDOW_MIN = 0 -> use standard windows)

Z_WINDOW_MAX

Floating point value representing the grid distance (m) indicating the top of the window to be specified and fixed for outputing dot plots.

(Default: X_WINDOW_MAX = 0 -> use standard windows)

Z_WINDOW_MIN

Floating point value representing the grid distance (m) indicating the bottom of the window to be specified and fixed for outputing dot plots.

III.B.9.b. ADPIC_METPARAMS namelist

DIFF_ON_X

Flag which allows user to turn diffusion off in *x* (east-west) direction. (Note: *x* is not downwind distance). Used for DIFF_METHOD = 'RDM' only.

(Default: DIFF_ON_X = 'Y')

DIFF_ON_Y

Flag which allows user to turn diffusion off in *y* (north-south) direction. (Note: *y* is not crosswind distance). Used for DIFF_METHOD = 'RDM' only.

(Default: DIFF_ON_Y = 'Y').

DIFF_ON_Z

Flag which allows user to turn diffusion off in *z* (vertical) direction. Used for DIFF_METHOD = 'RDM' only.

(Default: DIFF_ON_Z = 'Y')

GAUSS_DIFF_MULT_X

Floating point value used as a multiplier to QNSH, which determines how long an ADPIC particle remains a source particle. Also see GAUSS_DIFF_MULT_Z. Used only with DIFF_METHOD = 'GRADI'.

(Default: GAUSS_DIFF_MULT_X = 1.0)

GAUSS_DIFF_MULT_Z

Floating point value used as a multiplier to QNSV, which determines how long an ADPIC particle remains a source particle. Also see GAUSS_DIFF_MULT_X. Used only with DIFF_METHOD = 'GRADI'.

(Default: GAUSS_DIFF_MULT_Z = 1.0)

HORZ_TIME_SCALE

Horizontal turbulence time scale, τ_H (seconds). Applies to TURB_PARAM_VERT = 'KH_SIGTHETA_USER' option only, and must be input for that option. (Default: 50 s for TURB_PARAM_VERT = 'KH_SIGTHETA')

INTERP_VEL_WINDS

8-character string ('NONE', 'LINEAR1', 'LINEAR2', 'GAUSS1', 'GAUSS2') used to invoke time-interpolation of MEDIC or MATHEW wind fields. If the flag is absent from the namelist or set to 'NONE", ADPIC does no interpolation of wind fields and uses wind datasets in a stepwise manner. 'LINEARn' results in linear interpolation between wind field datasets at each model timestep, while

'GAUSSn' results in Gaussian interpolation. For n=1, ADPIC uses dataset start times to define the time intervals for interpolation; for n=2, ADPIC shifts each wind field starting time by half its duration time and uses these shifted times to define the interpolation time intervals. In other words, the winds are true at the beginning of the dataset period for n=1, and at the mid-point of the dataset period for n=2. If calculation time is outside the time range of available wind data, ADPIC uses the wind dataset nearest in time with no interpolation. Parameters in ADPIC which are not strong functions of wind velocities are computed at wind dataset intervals.

(Default: INTERP_VEL_WINDS = 'NONE')

INTERP_WINDS_STEP_MEAN

A character variable ('y'/'n') which allows a more accurate interpolation of winds at the mean position during the random vertical displacement. If 'y', a more accurate calculation is possible in instances where winds change rapidly in the vertical.

(Default: INTERP_WINDS_STEP_MEAN = 'N')

INV_OBUKHOV_LEN_ML

Array (max size = MIXTIM=50) of floating point values representing the inverse of the Monin-Obukhov length in the mixed layer. The Monin-Obukhov length (usually referred to as L) is the height above the surface at which mechanical and thermal turbulence are of equal importance. From an estimate of SFC_ROUGH_HGT and the atmospheric stability, INV_OBUKHOV_LEN_ML can be found from the nomogram in Figure III.B.2. INV_OBUKHOV_LEN_ML is a time-varying parameter with date/times controlled by MET_START_DATE / MET_START_TIME.

Warning errors:	INV_OBUKHOV_LEN_ML < -0.15
	INV_OBUKHOV_LEN_ML > 0.15
Fatal errors:	INV_OBUKHOV_LEN_ML < - 1.00
	INV_OBUKHOV_LEN_ML > 1.00
	INV_OBUKHOV_LEN_ML > 0.0 and STAB_CLASS < 4
	INV_OBUKHOV_LEN_ML < 0.0 and STAB_CLASS > 4

INV_OBUKHOV_LEN_SL

Array (max size = MIXTIM=50) of floating point values representing the inverse of the Monin-Obukhov length in the surface layer. The Monin-Obukhov length (usually referred to as L) is the height above the surface at which mechanical and thermal turbulence are of equal importance. From an estimate of SFC_ROUGH_HGT and the atmospheric stability, INV_OBUKHOV_LEN_SL can be found from the nomogram in Figure III.B.2.

Warning errors:	INV_OBUKHOV_LEN_SL < -0.15
	INV_OBUKHOV_LEN_SL > 0.15
Fatal errors:	INV_OBUKHOV_LEN_SL < - 1.00
	INV_OBUKHOV_LEN_SL > 1.00

INV_OBUKHOV_LEN_SL > 0.0 and STAB_CLASS < 4

INV_OBUKHOV_LEN_SL < 0.0 and STAB_CLASS > 4

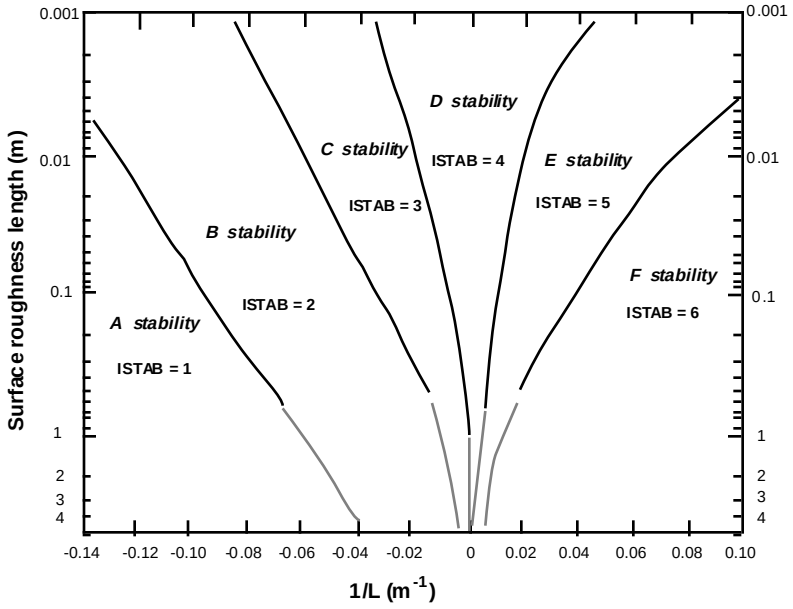


Figure III.B.2. Relation of Monin-Obukhov L to Pasquill class and roughness length (Golder, 1972)

K_VON_KARMAN

Optional input parameter to override default value of the von Karman constant, *k*. (Default: K_VON_KARMAN = 0.4)

KZ_CONSTANTZ

Constant K_z value used when TURB_PARAM_VERT= 'KZ_CONSTANTZ'. Must be input by the user. No internal default value.

KZ_LINEARZ_REF

Reference value K_{zref} (m^2/s) for linear vertical eddy diffusivity profile, $K_z = K_{z\text{ref}}\left(z/z_{\text{ref}}\right)$. Applies to TURB_PARAM_VERT = 'KZ_LINEARZ' option only, and must be input for that option.

KZ_LINEARZ_REF_HGT

Reference height z_{ref} (m) for power law vertical eddy diffusivity reference value. Applies to TURB_PARAM_VERT = 'KZ_LINEARZ' only, and must be input for that option.

KZ_SIMTHRY_PHI_A

Constant, *a*, for similarity theory vertical eddy diffusivity profile phi function. Applies to TURB_PARAM_VERT = 'KZ_SIMTHRY_USER' option only, and must be input for that option.

KZ_SIMTHRY_PHI_B

Constant, b, for similarity theory vertical eddy diffusivity profile phi function. Applies to TURB_PARAM_VERT = 'KZ_SIMTHRY_USER' option only, and must be input for that option.

KZ_SIMTHRY_C

Constant, c, for similarity theory vertical eddy diffusivity profile. Used only when TURB_PARAM_VERT = 'KZ_SIMTHRY_USER' option is used, and must be input for that option.

KZ_SIMTHRY_C2

Constant, c_2, for similarity theory vertical eddy diffusivity profile. Used only when TURB_PARAM_VERT = 'KZ_SIMTHRY_USER' option is used, and must be input for that option.

KZ_SIMTHRY_C3

Constant, c_3, for similarity theory vertical eddy diffusivity profile. Used only when TURB_PARAM_VERT = 'KZ_SIMTHRY_USER' option is used, and must be input for that option.

KZ_SIMTHRY_TROPO

Constant value of K_z^{tropo} used above $z > h$, but only when TURB_PARAM_VERT= 'KZ_SIMTHRY_USER' (refer to TURB_PARAM_VERT parameter for other constants that must also be specified to use the 'KZ_SIMTHRY_USER' option).

MAX_DELZ_SHIFT

Integer representing the maximum number of cells that a particle can pop up. There is a tendency for particles to impact with topography in complex terrain and become 'stuck'. In order to help alleviate this problem we have introduced a procedure which allows particles to pop up a given number of times to get the particles back out into the flow field. This procedure involves moving the particle up one cell at a time in the vertical until the particle has either cleared the topography or has reached the allowed limit of vertical shift (MAX_DELZ_SHIFT grid cells) or the upper limit of the grid is encountered. If this limit (MAX_DELZ_SHIFT) has been reached, the particle is moved only with the velocity component parallel to the topography. This procedure will allow the particle to either move over or around the topography. MAX_DELZ_SHIFT specifies the limit of cells a particle is allowed to be shifted vertically.

(Default: MAX_DELZ_SHIFT = 1)

MAX_DITHER

This parameter is used only for DIFF_METH = 'GRADI' (Gradient Diffusion method). It is a floating point value specifying the multiplicative factor of the timestep delta for a given computational cycle, and that product is then used as a time to multiply the maximum particle speed in each direction. When the advection component of the velocity is much less than the diffusion component, the particles will try to line up on grid lines. This is particularly true in the vertical direction. To help break up this alignment we have introduced a small random change (a dither,

similar to that in control systems). MAX_DITHER controls the amount of this small random movement. Since the resulting 'small' distance is in effect a fraction of the maximum movement for the next cycle. By then multiplying by a random number (between -1. and +1.) the correction acts to prevent the alignment on horizontal grid lines.

(Default: MAX_DITHER = 0.01)

MAX_GLOBAL_DELTA_T

A parameter which can be used to limit the global timestep to a specific value. A positive number representing a timestep (secs) is input if this feature is to be active. (No default)

MET_START_DATE

Array (max size = MAXTIM = 100) of 7-character strings specifying the dates (ddmmmyy) at which new values of the time-varying met parameters (TURB_PARAM_TYPE, STAB_CLASS, INV_OBUKHOV_LEN_ML, INV_OBUKHOV_LEN_SL, ML_HGT, SIGMA_THETA_FACT, SIGMA_THETA, SIGMA_THETA_HGT, and PRECIP_SCAVENG_COEF) take effect. Also see MET_START_TIME, which specifies the times.

MET_START_TIME

Array (max size = MAXTIM = 100) of 8-character strings specifying the times (hh:mm:ss) at which new values of the time-varying met parameters (TURB_PARAM_TYPE, STAB_CLASS, INV_OBUKHOV_LEN_ML, INV_OBUKHOV_LEN_SL, ML_HGT, SIGMA_THETA_FACT, SIGMA_THETA, SIGMA_THETA_HGT, and PRECIP_SCAVENG_COEF) take effect. Also see MET_START_DATE, which specifies the dates.

MIN_MASS_FRACT

Array (max size = MIXTIM=50) of floating point values representing the factor used to determine whether or not an ADPIC particle should be eliminated due to decreased mass or activity: If mass or activity is less than MIN_MASS_FRACT times the original mass or activity of the ADPIC particle, then the particle is eliminated from the problem.

(Default: MIN_MASS_FRACT = 0.001)

ML_HGT

Array (max size = MIXTIM=50) of floating point values representing the height (m) of the mixed layer above terrain. The height of the mixed layer is that demarcation between well mixed, turbid air below, and relatively clear air above. The mixed layer, typically ranging from 300 to 3000 m above terrain, basically indicates the height of the temperature inversion base. ML_HGT is usually obtained from the radiosonde nearest the source (or area of interest). In the absence of a temperature profile, observations have often shown that the wind will shear rapidly in direction and be accompanied by a speed minima as the sonde passes through the inversion layer.

Warning errors: ML_HGT $\leq$ 50.0

ML_HGT $\geq$ 3000.0

Fatal errors: ML_HGT < DELZ

ML_HGT $\geq$ 20000.0

PWRLAW_EXPNT_U

Exponent *m* for power law wind profile used to generate mean winds when. WIND_TYPE= 'POWER_LAW'. This allows mean winds to be generated without using input gridded winds. See parameters U_MEAN_DIR, U_MEAN_HGT, AND UMEAN_REF. Must be input if WIND_TYPE= 'POWER_LAW'.

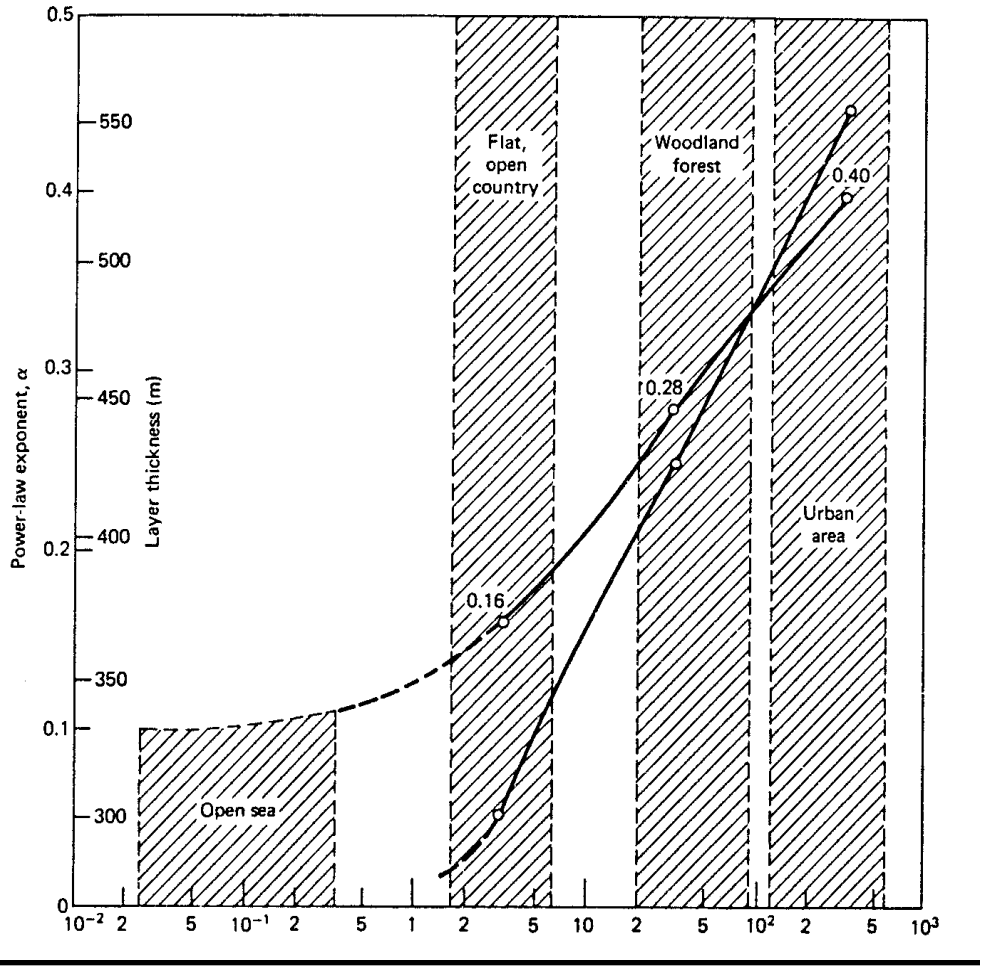
PWR_LAW_EXPNT_SL

Array (max size = MIXTIM=50) of floating point values representing the power law exponent wind speed in the surface layer. PWR_LAW_EXPNT_SL is used within the code to calculate the mean wind at the reference height U_MEAN_HGT for use in the calculation of friction velocity. It can be calculated directly from surface-layer wind observations in the vertical using the power law formula given by Eq. (II.C.1). Alternatively, it can be obtained as a function of surface roughness height and atmospheric stability, as shown in Fig. III.B.1.

Warning errors: PWR_LAW_EXPNT_SL > 0.5

Fatal errors: PWR_LAW_EXPNT_SL < 0.0

PWR_LAW_EXPNT_SL > 10.0



Surface roughness height (Zo)– *m*

The exponent in the power law and the height of the boundary layer as functions of the roughness height and the type of terrain. From Davenport (1965).

Z_o = surface roughness height $(\mathrm{m}) \sim h_o/30$

where h_o = average height of terrain features (e.g., building, trees, grass)

PWR_LAW_EXPNT_SL is obtained from graph using appropriate Z_o and the following correction for stability (if necessary):

Stable:	add 0.1
Neutral:	as is
Unstable:	subtract 0.1

Figure III.B.1 Modeling the planetary boundary layer

PRECIP_SCAVENG_COEF

Array (max size = MIXTIM=50) of floating point values representing the washout coefficient for rainout calculations. PRECIP_SCAVENG_COEF must be accompanied by entries for RAIN_START and RAIN_STOP if only the first element of PRECIP_SCAVENG_COEF is set. Otherwise, PRECIP_SCAVENG_COEF is controlled by date/times of the met_start_date/met_start_time arrays. Suggested values: PRECIP_SCAVENG_COEF = 1.0E-3 for light rain, 2.0E-3 for light snow.

RAIN_START

13-character string specifying the time (ddd::hh:mm:ss) relative to the problem start time (RUN_START_TIME) when the rain is to start. This is used in conjunction with RAIN_STOP (rain stop time) and PRECIP_SCAVENG_COEF (washout coefficient), first element only.

RAIN_STOP

13-character string specifying the time (ddd::hh:mm:ss) relative to the problem stop time (RUN_START_TIME) when the rain is to stop. This is used in conjunction with RAIN_START (rain start time) and PRECIP_SCAVENG_COEF (washout coefficient).

SFC_ROUGH_HGT

Floating point value representing the surface roughness height (m). The surface roughness height is approximately the average height of most land-based features (e.g., grass, trees, buildings) divided by 30. The analyst should consider features over the entire gridded area before entering a value. This is a subjective parameter since we are usually dealing with very diverse geographical areas. Figure III.B.3 gives typical values of surface roughness for various terrain types.

Warning errors:	SFC_ROUGH_HGT $\geq$ 2.0
Fatal errors:	SFC_ROUGH_HGT $\leq$ 0.0
	SFC_ROUGH_HGT > 5.0

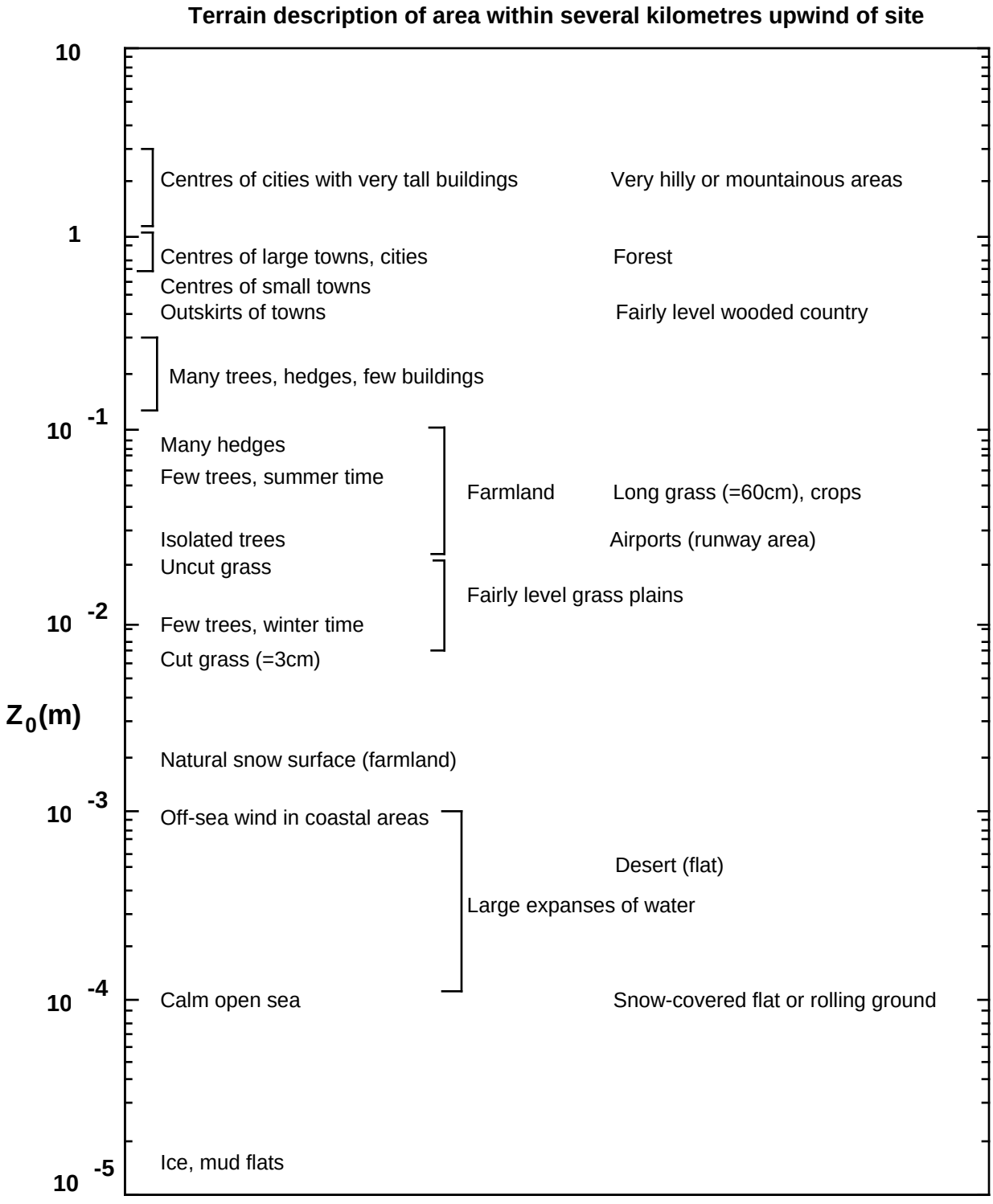


Figure III.B.3. z_0 values for typical terrain types (After ESDU 72026, 1972)

SIGH_PWRLAWX_ALPHA

Constant α in power-law equation for horizontal concentration standard deviation versus downwind distance, $\sigma_H = \alpha x^{\beta}$, used to calculate horizontal eddy diffusivity. Applies to TURB_PARAM_HORZ = 'SIGH_PWRLAWX_USER' option only, and must be input for that option.

SIGH_PWRLAWX_BETA

Constant β in power-law equation for horizontal concentration standard deviation versus downwind distance, $\sigma_H = \alpha x^{\beta}$, used to calculate horizontal eddy diffusivity. Applies to TURB_PARAM_HORZ = 'SIGH_PWRLAWX_USER' option only, and must be input for that option.

SIGH_PWRLAWX_XMAX

Maximum downwind distance which the power law equation for horizontal concentration standard deviation versus downwind distance is to be used. After this distance the value of horizontal diffusivity, K_H, is held constant at its value at this distance. Applies to TURB_PARAM_HORZ = 'SIGH_PWRLAWX_USER' option only, and must be input for that option.

SIGMA_THETA_FACT

Array (max size = MAXTIM = 100) of floating point values specifying a time-varying factor by which the sigma theta profile generated from the SIGMA_THETA array of the SIGMA_THETA_DATA namelist is multiplied.

SIGZ_PWRLAWX_GAMMA

Constant γ in power-law equation for vertical concentration standard deviation versus downwind distance, $\sigma_z = \gamma x^{\rho}$, used to calculate vertical eddy diffusivity. Applies to TURB_PARAM_VERT = 'SIGZ_PWRLAWX_USER' option only, and must be input for that option.

SIGZ_PWRLAWX_RHO

Constant ρ in power-law equation for vertical concentration standard deviation versus downwind distance, $\sigma_z = \gamma x^{\rho}$, used to calculate vertical eddy diffusivity. Applies to TURB_PARAM_VERT = 'SIGZ_PWRLAWX_USER' option only, and must be input for that option.

SIGZ_PWRLAWX_XMAX

Maximum downwind distance (m) which the power law equation for vertical concentration standard deviation versus downwind distance is to be used. After this distance the value of vertical diffusivity, K_z, is held constant at its value at this distance. Applies to TURB_PARAM_HORZ = 'SIGH_PWRLAWX_USER' option only, and must be input for that option.

STAB_CLASS

Array (max size = MIXTIM=50) of integers (1-6) representing the Pasquill-Gifford stability category, A-F. Stability class is used by some of the turbulence parameterization options (see

TURB_PARAM_VERT and TURB_PARAM_HORZ). If a horizontal turbulence parameterization based on sigma theta data is used, then it is also used to specify a default sigma theta surface value if no sigma theta data is input (see SIGMA_THETA_DATA namelist). There are many ways to determine stability class. One is the Pasquill-Turner method, which incorporates insolation assessments based on hourly cloud observations. The heat flux at the earth's surface is qualitatively estimated from these cloud observations, and then related to stability on the basis of the station wind speed. This method is similar to the method used to define the stability categories during the original development of the Pasquill-Gifford curves. STAB_CLASS must be representive of the entire model domain since only one value at a time is used for the entire domain.

Warning errors: NONE
Fatal errors: STAB_CLASS < 1
STAB_CLASS > 6

TIMESTEP_GLOBAL_FRACT

A parameter which can be used to decrease the timestep for both the Gradient Diffusion and RDM options. TIMESTEP_GLOBAL_FRACT is a decimal fraction multiplier applied after the timestep after it is determined in the code. For the DIFF_METH='RDM' option, it applies only to the global time step and not the local particle time step (see TIME_STEP_LOCAL_FRACT). Allowed values are 0. to 1.

(Default: TIMESTEP_GLOBAL_FRACT = 1.0)

TIMESTEP_LOCAL_MAX

A parameter (secs)which can be used to provide an upper limit for the local (particle-specific) timestep used to calculate particle movement. This parameter is likely used only in sensitivity studies of the RDM calculation scheme. Used for DIFF_METHOD = 'RDM' only. (Default: 'infinity')

TIMESTEP_LOCAL_MIN

A parameter (secs)which can be used to provide a lower limit for the local (particle-specific) timestep. This parameter is likely used only in sensitivity studies of the RDM calculation scheme. Used for DIFF_METHOD = 'RDM' only. (Default: 0.0)

TIMESTEP_LOCAL_FRACT

A parameter (decimal fraction) which multiplies the internally calculated local (particle-specific) timestep, scaling it either upward or downward. If timestep_local_min and/or timestep_local_max are set, those limits have precedence over a timestep determined by scaling from this variable (timestep_local_fract). Used for DIFF_METHOD = 'RDM' only. (Default: 1.0)

TURB_PARAM_TYPE

Array (max size = MIXTIM=50) of integers (1-5) specifying the turbulence parameterization to be used. Current possible values of TURB_PARAM_TYPE for use with gradient diffusion are 1, 2, 4, and 5. Used only with DIFF_METHOD = 'GRADI'.

When stack plume rise is used, the input parameters for the inverse Monin-Obukhov length (INV_OBUKHOV_LEN_SL) and the top of the mixed layer (ML_HGT) are required for computation of friction velocity for all values of TURB_PARAM_TYPE.

TURB_PARAM_TYPE	Turbulence parameterization to be used
1	Gaussian coefficients, Pasquill-Gifford
2	Gaussian coefficients, Idaho Falls MesoDif model

The following statements apply to both TURB_PARAM_TYPE's 4 and 5: Generalized turbulence parameters are computed for the entire atmospheric boundary layer as a function of wind profile, stability, surface roughness, and mixing layer thickness. The following input parameters must be provided in the Adpic.nml file: stability category (STAB_CLASS), inverse Monin-Obukhov length (INV_OBUKHOV_LEN_SL), top of mixing layer (ML_HGT), power law exponent in surface layer (PWR_LAW_EXPNT_SL), and surface roughness height (SFC_ROUGH_HGT). The vertical turbulence coefficient Kz is from Businger, Arya, and Deardorff in the outer mixing layer, and similarity theory in the surface layer. The Idaho Falls MesoDif Gaussian diffusion sigma-y curves are used for sub-grid diffusion. These coefficients result in more diffusion under stable conditions due to meandering effects when compared to the classical Pasquill-Gifford sigma-y curves.

TURB_PARAM_TYPE = 4 → Sigma-Theta method. Kh's obtained from Taylor's Theory using sigma-theta profile (fluctuation of the horizontal wind direction). The sigma-theta profile is input via the SIGMA_THETA and SIGMA_THETA_HGT arrays in the namelist sigma_theta_data of the ADPIC.NML namelist file. This diffusion scheme is suitable for continuous and moderately elevated releases within the mixed layer where sigma-theta data are available.

TURB_PARAM_TYPE = 5 → Energy dissipation rate (epsilon) method. Kh's obtained from scale-dependent horizontal diffusion using energy dissipation rate, epsilon. This diffusion scheme is suitable for instantaneous surface and elevated releases where horizontal diffusion is based upon the atmospheric energy dissipation rate.

TURB_PARAM_VERT

Array (max size = MIXTIM=50) of 22-character strings identifying *vertical* turbulence parameterization option to be used. The parameter applies only if DIFF_METHOD = 'RDM' in the ADPIC_CONTROL namelist. The recommended value is TURB_PARAM_VERT= 'KZ_SIMTHRY'. Possible values are as follows:

TURB_PARAM_VERT	Description
'KZ_SIMTHRY'	K_z as function of z calculated from boundary layer similarity theory as function of u_*, h, z and L . A constant value of K_z^{tropo} is used above $z = h$..
'KZ_SIMTHRY_USER'	$K_z(z)$ as function of z calculated from boundary layer similarity theory as function of u_*, h, z and L, but with *user-specified* coefficients a, b, c , and K_z(tropo (KZ_SIMTHRY_PHI_A, KZ_SIMTHRY_PHI_B, KZ_SIMTHRY_C, KZ_SIMTHRY_C2, KZ_SIMTHRY_C3 and KZ_SIMTHRY_TROPO *must be input* to use this option)
'SIGZ_PWRLAWX_PG'	K_z as function of downwind distance, x, calculated from *Pasquill-Gifford* $\sigma_z(x)$ in a power law form.
'SIGZ_PWRLAWX_USER'	K_z as function of downwind distance x (in meters) calculated from $\sigma_z(x)$ (in meters) using *user-specified* power law coefficients γ and ρ (SIGZ_PWRLAWX_GAMMA and SIGZ_PWRLAWX_RHO) in the power law equation: $\sigma_z = \gamma x^{\rho}$.
'KZ_LINEARZ'	K_z calculated from power law equation, $K_z = K_{z\text{ref}}\left(z/z_{\text{ref}}\right)$, where K_{zref} and z_{ref}. (KZ_LINEARZ_REF and KZ_LINEARZ_REF_HGT) are specified by the user.
'KZ_CONSTANTZ'	K_z fixed at constant value = KZ_CONSTANTZ, specified by the user.

TURB_PARAM_HORZ

Array (max size = MIXTIM=50) of 22-character strings identifying *horizontal* turbulence parameterization option to be used. This parameter applies only if DIFF_METHOD = 'RDM'. The recommened value is TURB_PARAM_VERT= 'SIGH_SIGTHETA'. Possible values are as follows:

TURB_PARAM_HORZ	Description
'SIGH_SIGTHETA'	K_H as function of t calculated from Draxler σ_y (t) using input σ_θ data (see SIGMA_THETA_DATA namelist) and internal-default value of HORZ_TIME_SCALE, τ_H.
'SIGH_SIGTHETA_USER'	K_H as function of t calculated from Draxler $\sigma_y(t)$ using input σ_θ data (see SIGMA_THETA_DATA namelist) and *user-input* value of HORZ_TIME_SCALE, τ_H.
'SIGH_PWRLAWX_PG'	K_H as function of downwind distance x calculated from *Pasquill-Gifford* $\sigma_y(x)$ in a power law form ($\sigma_H = \sigma_y = \alpha x^{\beta}$)

'SIGZ_PWRLAWX_USER'	K_z as function of downwind distance *x* (in meters) calculated from $\sigma_z(x)$ (in meters) using *user-specified* power law coefficients γ and ρ (SIGZ_PWRLAWX_GAMMA and SIGZ_PWRLAWX_RHO) in the power law equation: $\sigma_z = \gamma x^{\rho}$.
'SIGH_PWRLAWX_MESO'	K_H as function of downwind distance *x* calculated from *MESODIF* $\sigma_y(x)$
'SIGH_PWRLAWX_LONGRANGE'	K_H as function of downwind distance *x* calculated from *Dan Rodriguez's long-range* $\sigma_y(x)$
'SIGH_PWRLAWX_USER'	K_H as function of downwind distance*x* (in meters) calculated from $\sigma_y(x)$ (in meters) using *user-specified* power law coefficients SIGH_PWRLAWX_ALPHA, α, and SIGH_PWRLAWX_BETA, β in the power equation: $\sigma_H = \alpha x^{\beta}$.

U_CALM

Floating point value representing the minimum value (m/s) for wind speed in friction velocity calculations. The friction velocity is computed at every horizontal grid point, and is dependent on the surface rougness height, stability, and wind speed at the grid points. In order to prevent the friction velocity from becoming zero, U_CALM is used for the wind speed when the wind speed is zero.

(Default: U_CALM = 0.2)

U_MEAN_DIR

Mean horizontal wind direction (in degrees) using standard meteorolgical convention (clockwise from north for direction wind is blowing from). Applies to WIND_TYPE= 'POWER_LAW' option only, and must be input for that option.

U_MEAN_HGT

Floating point value representing the height (m) of mean wind used to compute friction velocity.

(Default: U_MEAN_HGT = 10.0)

U_MEAN_REF

Reference wind speed value $\bar{u}_1$ (m/s) for power law wind profile:

$$\bar{u}(z) = \bar{u}_1 \left(\frac{z}{z_1} \right)^m .$$

Applies to WIND_TYPE= 'POWER_LAW' option only, and must be input for that option.

U_MEAN_REF_HGT

Reference height z_1 (m) for power law wind profile reference wind speed, U_MEAN_REF. Applies to WIND_TYPE= 'POWER_LAW' option only, and must be input for that option.

III.B.9.c. ADPIC_SOURCE namelist

AMBIENT_TEMP

2-D array (MAXTIM = 100, MAXSOURC = 9) of floating point values representing the ambient air temperature (K). as a function of time, for each source. This array is part of the stack plumerise model. If AMBIENT_TEMP and STACK_TEMP are specified, then HEAT_EMISSION_RATE should be omitted.

(Default: AMBIENT_TEMP = 288.)

CENTER_HGT

2-D array (MAXTIM = 100, MAXSOURC = 9) of floating point values representing the height (m) above terrain of the source center.

CHEM_BIO_DECOMP_TAU

Array (MAXTIM = 100, MAXSOURC = 9) of time-varying real values specifying the time (hours) for decomposition of a chemical/biological material to 1/e of the original amount. Usually a decay rate in percent per minute is known (X %/min), in which case CHEM_BIO_DECOMP_TAU = 100 / (X*60min/hr) = 1.667 / X. This parameter will be used if DECAY_MODE = 'CHEM_BIO_DECOMP'.

(Default: none -- this is a required parameter for DECAY_MODE = 'CHEM_BIO_DECOMP')

DECAY_MODE

Array (MAXSOURC = 9) of 16-Character strings ('STABLE', 'RAD_DECAY', 'CHAIN_RAD_DECAY', 'CHEM_BIO_DECOMP') specifying if and how the source material will decrease in activity or amount, or change its form, state, or other characteristics. 'STABLE' implies that the material will not change. 'RAD_DECAY' implies that the material will undergo a simple radioactive decay (used in conjunction with the HALFLIFE parameter), which will not include any consideration of daughter products. 'CHAIN_RAD_DECAY' triggers the hybrid particle capability (activating the use of the ADPIC_RADPARAMS namelist), which will allow for complex radioactive decay, including daughter products. 'CHEM_BIO_DECOMP' implies that the material will undergo a chemical or biological time-dependent decomposition (see CHEM_BIO_DECOMP_TAU parameter).

(Default: DECAY_MODE = 'RAD_DECAY' = simplistic radioactive decay used)

EXIT_VEL

Array (max size = MAXSOURC = 9) of floating point values specifying the vertical exit velocity (m/s) of the gas emitted from the stack of each ADPIC source. Used in the plume rise calculation of buoyancy flux and momentum flux. Normal range of values are 3 to 50 m/sec. Any value less than or equal to 2 m/sec is treated as zero. The ratio of horizontal wind velocity at the source location to the vertical exit velocity is used to determine if vertical plume equations or bent-over plume equations are used. If that ratio is less than 0.1, then vertical plume equations are used.

HALFLIFE

Array (max size = MAXSOURC = 9) of floating point values specifying the radioactive half-life (hours) of each ADPIC source. Use HALFLIFE = 0 for all non-radioactive substances.

HEAT_EMISSION_RATE

Array (max size = MAXSOURC = 9) of floating point values specifying the sensible heat emission rate (MW) for each ADPIC source. This is an optional parameter used (only if STACK_TEMP and AMBIENT_TEMP is not input) to calculate buoyancy flux in plume rise calculations. Typical values might be 0.01 to 100 MW. This array is part of the stack plumerise model.

INV_HGT_AT_STACK

2-D array (MAXTIM = 100, MAXSOURC = 9) of floating point values representing the height (m) of strong inversion affecting plume rise. This is an optional parameter; if a value is input, plume rise for source will be limited by this height, but only in cases involving buoyancy plume rise and neutral/unstable stability classes. If a value is not input, the ML_HGT value serves this function, but only for cases of vertical plumes in neutral/unstable stability classes.

PART_DENSITY

Array (max size = MAXSOURC = 9) of floating point values specifying the particle density (kg/m3) for each ADPIC source. PART_DENSITY is usually a value that has been normalized to 1 g/cc = 1000 kg/m3, but the actual particle density may also be used. Note that the PART_DENSITY chosen should be consistent with the values for PART_DIAM_MEDIAN, PART_DIAM_MIN, and PART_DIAM_MAX.

(Default: PART_DENSITY = 1000. $\rightarrow$ 1 g/cc)

PART_DIAM_MAX

Array (max size = MAXSOURCE = 9) of floating point values specifying the largest particle diameter (μm) of the effluent. It is usually taken to be two orders of magnitude greater than PART_DIAM_MEDIAN. The terminal fall velocities for those particles falling outside the realm of validity of Stoke's Law, i.e., Reynolds number greater than one, are calculated using an algorithm derived from McDonald's computational aid for the terminal fall velocity of spheres. See Section

II.B.4.d for more information. If PART_DIAM_MEDIAN is set to zero, no entry is required for PART_DIAM_MAX. Typical values are:

PART_DIAM_MAX = 200.0 → Pu HE detonation, based upon CS-I data

PART_DIAM_MAX = 200.0 → Pu fire, based upon Rocky Flats data

PART_DIAM_MEDIAN

Array (max size = MAXSOURCE = 9) of floating point values specifying the effluent particle size median diameter (μm) for each ADPIC source. If there is no particular range of particle sizes, use PART_DIAM_MEDIAN of zero, which will result in a zero gravitational settling velocity. Typical values are:

PART_DIAM_MEDIAN = 40.0 → Pu HE detonation, based upon CS-I data

PART_DIAM_MEDIAN = 0.6 → Pu fire, based upon Rocky Flats data

PART_DIAM_MEDIAN = 0.0 → passive pollutant (gases, smoke,...)

PART_DIAM_MIN

Array (max size = MAXSOURCE = 9) of floating point values specifying the smallest particle diameter (μm) of the effluent. It is usually taken to be two orders of magnitude less than PART_DIAM_MEDIAN. If PART_DIAM_MEDIAN is set to zero, no entry is required for PART_DIAM_MIN. Typical values are:

PART_DIAM_MIN = 0.2 → Pu HE detonation, based upon CS-I data

PART_DIAM_MIN = 0.2 → Pu fire, based upon Rocky Flats data

PART_DIAM_SGD

Array (max size = MAXSOURCE = 9) of floating point values specifying the standard geometric deviation of the log-normal particle size distribution for each ADPIC source, and is defined as the following ratio:

$$\frac{\text{radius at 50\%}}{\text{radius at 16\%}} \text{ or } \frac{\text{radius at 84\%}}{\text{radius at 50\%}}$$

on a log-normal frequency distribution plot. If PART_DIAM_MEDIAN is set to zero, no entry is required for PART_DIAM_SGD. Typical values are:

PART_DIAM_SGD = 5.71 → Pu HE detonation, based upon CS-I data

PART_DIAM_SGD = 3.0 → Pu fire, based upon Rocky Flats data

PART_GEN_MODE

Array (max size = MAXSOURCE = 9) of 20-character strings ('MASS_ACTIVITY_FRACT' or 'TIME_FRACT') used to control whether the number of marker particles for an ADPIC source

generated for a particular time interval is proportional to the activity/mass of the source, or proportional to the time that the source is "on". Although the first method is generally recommended, the user should be aware that while the particle statistics will be improved during peak release times, they will also be degraded during the non-peak release times.

(Default: PART_GEN_MODE = 'MASS_ACTIVITY_FRACT')

RELEASE_START (recommend use of SOURCE_START_DATE and SOURCE_START_TIME)

Array (max size = MAXSOURCE = 9) of 13-character time interval strings (ddd::'hh:mm:ss) specifying the release start time relative to the run start time (RUN_START_DATE \ RUN_START_TIME for non-restart, RESTART_DATE \ RESTART_TIME for restart) for each ADPIC source. Use of SOURCE_START_DATE and SOURCE_START_TIME will override use of RELEASE_START. Use of SOURCE_START_DATE and SOURCE_START_TIME instead of RELEASE_START is recommended because they are absolute times and allow for time-varying source rates.

RELEASE_STOP (recommend use of SOURCE_START_DATE and SOURCE_STOP_TIME)

Array (max size = MAXSOURCE = 9) of 13-character time interval strings(ddd::hh:mm:ss) specifying the release stop time relative to the run start time (RUN_START_DATE \ RUN_START_TIME for non-restart, RESTART_DATE \ RESTART_TIME for restart) for each ADPIC source. RELEASE_STOP will override RUN_DURATION. Use of SOURCE_START_DATE and SOURCE_START_TIME will override use of RELEASE_STOP. Use of SOURCE_START_DATE and SOURCE_START_TIME instead of RELEASE_STOP is recommended because they are absolute times and allow for time-varying source rates.

SFC_DEP_HGT

Floating point value representing the height (m) above ground level used for dry deposition calculations. SFC_DEP_VEL decreases linearly with height and goes to zero at the height specified by SFC_DEP_HGT.

(Default: SFC_DEP_HGT = DELZ from TOPOG.NML)

SFC_DEP_VEL

Array (max size = MAXSOURCE = 9) of floating point values specifying the surface deposition velocity (m/s) of the effluent. This is not a gravitational settling velocity, but rather a measure of the affinity of the material for the ground. SFC_DEP_VEL is a function of particle size, atmospheric stability, and surface roughness height. The surface deposition velocity is the downward particle flux divided by the particle concentration--so the units are m/sec. SFC_DEP_VEL is only used in the bottom grid cell to deposit particles onto the ground, and is weighted according to the distance from the bottom of the cell (i.e., zero at the top of the cell to full value at the bottom). The only mechanisms by which particles or a portion of the particle activity can be removed from the

atmosphere and deposited on the ground are via SFC_DEP_VEL, rainout, and gravitational settling. Experimental data have shown that SFC_DEP_VEL usually lies between 0.001 and 0.05 m/s:

Particulate Effluent	SFC_DEP_VEL (m/s)
Pu-238, Pu-239	0.001- 0.01
SO2, Ru	0.01- 0.03 (0.028)
Cs-137, Sr-90	0.001- 0.002
I-131	0.003
Inert particles	0.001- 0.002
HTO	0.005- 0.008

SIGX

Array (max size = MAXSOURC = 9) of floating point values specifying the horizontal standard deviation (m) in the X-direction (East-West) of the initial stabilized source geometry for each ADPIC source.

SIGY

Array (max size = MAXSOURC = 9) of floating point values specifying the horizontal standard deviation (m) in the Y-direction (North-South) of the initial stabilized source geometry for each ADPIC source.

SIGZ

Array (max size = MAXSOURC = 9) of floating point values specifying the vertical standard deviation (m) of the initial stabilized source geometry for each ADPIC source.

SOURCE_RATE

2-D array (MAXTIM = 100, MAXSOURC = 9) specifying the floating point values for the source rate. The actual units of the bin contents to be plotted by PLCNT are determined by a combination of SOURCE_RATE, DCON (a PLCNT parameter), and the ADPIC internal units. See the description for DCON in the PLCNT section of this User's Guide for a more complete discussion of the interrelation of these parameters.

SOURCE_START_DATE

Array (max size = MAXTIM = 100) of 7-character date strings (ddMMMyy) specifying the dates at which new values of the time-varying source parameters (SOURCE_RATE. CENTER_HGT, AMBIENT_TEMP, INV_HGT_AT_STACK, and VERT_TEMP_GRAD) take effect. Also see SOURCE_START_TIME, which specifies the times

SOURCE_START_TIME

Array (max size = MAXTIM = 100) of 8-character time strings (hh:mm:ss) specifying the times at which new values of the time-varying source parameters (SOURCE_RATE. CENTER_HGT,

AMBIENT_TEMP, INV_HGT_AT_STACK, and VERT_TEMP_GRAD) take effect. Also see SOURCE_START_DATE, which specifies the dates.

SPECIES

Array (max size = MAXSOURC = 9) of 10-character strings specifying the name of the effluent for each ADPIC source.

STACK_RADIUS

Array (max size = MAXSOURC = 9) of floating point values specifying the internal stack radius (m) for each ADPIC source. This is used to calculate buoyancy flux and momentum flux for plume rise calculations. It is NOT used to specify the initial source geometry. The parameters ZPOS_CUTOFF, ZNEG_CUTOFF, XPOS_CUTOFF, XNEG_CUTOFF, YPOS_CUTOFF, YNEG_CUTOFF, SIGX, SIGY, and SIGZ are always used for this and should be consistent with STACK_RADIUS. Typical values are 1 to 5 m.

STACK_TEMP

Array (max size = MAXSOURC = 9) of values specifying the temperature (degrees K) of the gas emitted from the stack of each ADPIC source. Used to calculate buoyancy flux for plume rise calculations. If STACK_TEMP and AMBIENT_TEMP are specified, then HEAT_EMISSION_RATE should be omitted.

TOTPART

Array (max size = MAXSOURC = 9) of integers specifying the total number of particles to be generated for each ADPIC source. TOTPART is used to calculate the particle release rate. Although ADPIC allows a total maximum of 20,000 particles from all sources at any given time, a continuous release with particle depletion (via deposition, advection, and upward diffusion) can easily release more than 20,000 particles without exceeding this limit at any one time. However, if this maximum is exceeded, ADPIC will redistribute the mass or activity evenly among the 20,000 particles. Particles will be removed from the grid via deposition, advection, or upward diffusion.

(Recommendation: TOTPART = 3000 for multi-species, TOTPART = 5000 for single species)

USE_PLUME_RISE

Array (max size = MAXSOURC = 9) of integer flags ('Y' or 'N') indicating whether plume rise should be calculated for a specific source. If other plume rise parameters are present, but USE_PLUME_RISE = 'N', then the other parameters will be ignored.

(Default: USE_PLUME_RISE = 'N' → no plume rise)

VERT_TEMP_GRAD

Array (MAXTIM = 100, MAXSOURC = 9) of floating point values representing vertical temperature gradient (K/m) for use in plume rise calculations. The time varying dimension

corresponds to the same index in SOURCE_START_DATE and SOURCE_START_TIME. Used to calculate the Brunt-Vaisalla frequency. Defaults are provided in ADPIC for each stability class if a value is not input. (Note: The adiabatic, neutral temperature gradient is -0.0098 K/m. The standard atmosphere temperature gradient is -0.0065 K/m)

XLOC

Array (max size = MAXSOURC = 9) of floating point values specifying the Model x coordinate (km) of the source center. Note: Both XLOC and YLOC may not be closer than 3 grid cells from the edges of the ADPIC grid.

XNEG_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative X-direction (i.e., to the West or "left"), referenced to the source center (XLOC) for each ADPIC source.

(Note: since this is distance is reference to the source center, XNEG_CUTOFF should have a negative value.)

XPOS_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive X-direction (i.e., to the East or "right"), referenced to the source center (XLOC) for each ADPIC source.

YLOC

Array (max size = MAXSOURC = 9) of floating point values specifying the Model y coordinate (km) of the source center. Note: Both XLOC and YLOC must be no closer than 3 grid cells from the edges of the ADPIC grid.

YNEG_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative Y-direction (i.e., to the South or "left"), referenced to the source center (YLOC) for each ADPIC source.

(Note: since this is distance is reference to the source center, YNEG_CUTOFF should have a negative value.)

YPOS_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive Y-direction (i.e., to the North or "right"), referenced to the source center (YLOC) for each ADPIC source.

ZNEG_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative Z-direction (i.e., downward or to the “bottom”), referenced to the source center height (CENTER_HGT) for each ADPIC source.

(Note: since this is distance is reference to the source center height, ZNEG_CUTOFF should have a negative value.)

ZPOS_CUTOFF

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive Z-direction (i.e., upward or to the “top”), referenced to the source center height (CENTER_HGT) for each ADPIC source.

III.B.9.d. ADPIC_RADPARAMS namelist

The section ADPIC_RADPARAMS controls the Hybrid Particle part of Adpic. This permits the user to incorporate all nuclides that may be present in a given scenario. In the Hybrid Particle model, all nuclides that have similar properties may be grouped together and treated as a single source (an example would be noble gases). The time dependence of the activity for each source of hybrid particles is calculated in tables at the beginning of the problem and applied during the calculation The presence of radioactive decay products (daughters) is included in the calculation. One may remove the effect of any member of a decay chain by setting the dose conversion factor to zero.

DCF

A one-dimensional array (max size = MAX_DCF_LIST = 500) of real numbers. The dose conversion factor related to a specific DCF_NUCLIDE_NAME and DCF_PATH_DESCRIPTOR entry.

(Required parameter - no default)

DCF_NUCLIDE_NAME

A one-dimensional array (max size = MAX_DCF_LIST = 500) of 20-character strings (same format as the NUCLIDE_NAME) giving the names of the nuclides and their daughters. One entry for each dose conversion path.

(Required parameter - no default)

DCF_PATH_DESCRIPTOR

A one-dimensional array (max size = MAX_DCF_LIST = 500) of 20-character strings (same format as DOSE_PATH_DESCRIPTOR) giving the dose path for each DCF_NUCLIDE_NAME entry.

(Required parameter - no default)

DECAY_START_INTERVAL

A one-dimensional array (max size = MAXSOURC = 9) of 13-character strings (ddd::hh:mm:ss) that specify the time interval between initial decay time and initial release time. This is used to model cases where the course is initially contained and at a later time escapes.

(Default: DECAY_START_INTERVAL = 000::00:00:00)

DOSE_PATH_DESCRIPTOR

A one -dimensional array (max size = MAXSBIN = 30) of 20-character strings used to describe the dose path of the hybrid particle. Any character string may be used, but the ones used by model parameter are given below:

ACTIVITY - dose conversion factor is one (such as Deposition)

50YRCOINHALEADEFFECTI - 50 year committed dose equivalent by inhale.

50YRCOINHALADTHYROID - Thyroid dose 50 yr committed by inhal.

GAIRIMMADEFFECTIVEWB - Whole body gamma dose from air immersion

(Required parameter - no default)

NUMBER_OF_TOXIC_TIMES

Integer (max = 500) the number of entries in the hybrid_toxicity array. This should be large enough so the time interval between entries is small compared to the toxicity decay rates. Normally 100-200 should do. Larger number are fine if they don't noticeable slow the problem run speed.

(Default: NUMBER_OF_TOXIC_TIMES = 200)

NUCLIDE_MIX

A two-dimensional array (MAX_PARENTS = 100, MAXSOURC = 9) of real numbers. For each source, the mixes are normalized in the code to sum over the nuclide names to unity. This means that for each source, it is only the ratio of the nuclide_mixes that matters.

(Required parameter - no default)

NUCLIDE_NAME

A two-dimensional array (MAX_PARENTS = 100, MAXSOURC = 9) of 20-character strings that are the name of each parent nuclide for each source. Names have the form: element symbol-isotope number e.g. Cs-137, Sm-151.

(Required parameter - no default)

III.B.9.e. ADPIC_RECEPTOR namelist

RCPTR_DELX

Array (MAX_RCPTR = 500) specifying the x-offset value (real) from each receptor's reference trajectory (specified by TRAJ_X, TRAJ_Y, TRAJ_Z).

(Default: RCPTR_DELX = 500*0.0 = x offset from all receptors' specified trajectory coordinates is zero)

RCPTR_DELY

Array (MAX_RCPTR = 500) specifying the y-offset value (real) from each receptor's reference trajectory (specified by TRAJ_X, TRAJ_Y, TRAJ_Z).

(Default: RCPTR_DELY = 500*0.0 = y offset from all receptors' specified trajectory coordinates is zero)

RCPTR_DELZ

Array (MAX_RCPTR = 500) specifying the z-offset value (real) from each receptor's reference trajectory (specified by TRAJ_X, TRAJ_Y, TRAJ_Z).

(Default: RCPTR_DELZ = 500*0.0 = z offset from all receptors' specified trajectory coordinates is zero)

RCPTR_HGT_REF

Array (MAX_TRAJ = 500) of 12-character strings ('TERRAIN', 'SEA_LEVEL', or 'GRID_BOTTOM') specifying the z-value reference type.

(Default: RCPTR_HGT_REF = 'TERRAIN' = TRAJ_Z values are meters above ground)

RCPTR_ID

Array (MAX_RCPTR = 500) of 12-character strings specifying the identifiers for receptors.

(Default: blank string)

RCPTR_INFL_X_FRACT

Real value specifying the range of influence (expressed as a fraction of the advection grid cell size in the x-direction) of ADPIC particles on the receptors. If the grid cell size is large, and you just don't want any particles farther away than 0.25*DELX from each receptor location to contribute to the receptor concentration, you would set RCPTR_INFL_X_FRACT = 0.25.

(Default: RCPTR_INFL_X_FRACT = 1.0 = particles up to 1*DELX away contribute to the receptor concentrations)

RCPTR_INFL_Y_FRACT

Real value specifying the range of influence (expressed as a fraction of the advection grid cell size in the y-direction) of ADPIC particles on the receptors. If the grid cell size is large, and you just don't want any particles farther away than 0.25*DELY from each receptor location to contribute to the receptor concentration, you would set RCPTR_INFL_Y_FRACT = 0.25.

(Default: RCPTR_INFL_Y_FRACT = 1.0 = particles up to 1*DELY away contribute to the receptor concentrations)

RCPTR_INFL_Z_FRACT

Real value specifying the range of influence (expressed as a fraction of the advection grid cell size in the z-direction) of ADPIC particles on the receptors. If the grid cell size is large, and you just don't want any particles farther away than 0.25*DELZ from each receptor location to contribute to the receptor concentration, you would set RCPTR_INFL_Z_FRACT = 0.25.

(Default: RCPTR_INFL_Z_FRACT = 1.0 = particles up to 1*DELZ away contribute to the receptor concentrations)

RCPTR_INTERVAL

13-Character string (ddd::hh:mm:ss) specifying the accumulation period and integration interval for the moving receptor output.

(Default: RCPTR_INTERVAL = '000::01:00:00' = 1 hour)

RCPTR_MULT_GLOBAL

Real value specifying a multiplier for output concentration values used whenever not overridden by a group (via RCPTR_MULT_GROUP) or individual value (via RCPTR_MULT_IND).

(Default: RCPTR_MULT_GLOBAL = 1.0 = output concentration values unchanged)

RCPTR_MULT_GROUP

Array (MAX_TRAJ = 500) of real values specifying multipliers for output concentration values when not overridden by an individual value (via RCPTR_MULT_IND). (RCPTR_MULT_GROUP overrides RCPTR_MULT_GLOBAL)

(Default: none; optional parameter)

RCPTR_MULT_IND

Array (MAX_RCPTR = 500) of real values specifying multipliers for output concentration values. (RCPTR_MULT_IND overrides both RCPTR_MULT_GROUP and RCPTR_MULT_GLOBAL)

(Default: none; optional parameter)

RCPTR_SBIN

Array (MAX_RCPTR = 500) of integers specifying which ADPIC sampling bin is to be used for each receptor.

(Default: none; required parameter)

RCPTR_TITLE

70-Character string specifying the title line to appear in the RECEPTOR.DAT file.

(Default: blank string)

RCPTR_TRAJ

Array (MAX_RCPTR = 500) of integers specifying which reference trajectory (specified by TRAJ_X, TRAJ_Y, TRAJ_Z) is to be used for a particular receptor. For example, RCPTR_TRAJ = 2 5 4, would imply that trajectory #2 is to be used for receptor #1, trajectory #5 is to be used for receptor #2, and trajectory #4 is to be used for receptor #3.

(Default: none; required parameter)

TRAJ_X

Array (MAX_TRAJ_TIMES = 100, MAX_TRAJ = 500) of real values specifying the time-varying x-coordinates (km) of each trajectory.

(Default: none; required parameter)

TRAJ_Y

Array (MAX_TRAJ_TIMES = 100, MAX_TRAJ = 500) of real values specifying the time-varying y-coordinates (km) of each trajectory.

(Default: none; required parameter)

TRAJ_Z

Array (MAX_TRAJ_TIMES = 100, MAX_TRAJ = 500) of real values specifying the time-varying z-coordinates (m) of each trajectory.

(Default: none; required parameter)

TRAJ_DATE

Array (MAX_TRAJ_TIMES = 100) of 7-character strings specifying the date ('yymmmdd') portion of the date/time at which new values of the time-varying trajectory coordinate parameters (TRAJ_X, TRAJ_Y, TRAJ_Z) take effect.

(Default: none; required parameter)

TRAJ_TIME

Array (MAX_TRAJ_TIMES = 100) of 8-character strings specifying the time ('hh:mm:ss)' portion of the date/time at which new values of the time-varying trajectory coordinate parameters (TRAJ_X, TRAJ_Y, TRAJ_Z) take effect.

(Default: none; required parameter)

III.B.9.f. SIGMA_THETA_DATA namelist

SIGMA_THETA

Array (MAXTIM = 100 by MAXLEVEL_ALL = 35) of floating point values specifying sigma theta values (the standard deviation of wind direction, expressed in degrees). The first index matches entries in MET_START_DATE/MET_START_TIME while the last index matches the corresponding index of the array SIGMA_THETA_HGT. Sigma_theta is only used for TURB_PARAM_TYPE = 4, which selects the k-theory model. The scheme for determining the values of SIGMA_THETA depending upon the quantity of input data is given in the table below:

<table>
<tr><td></td><td colspan="3">SIGMA_THETA</td></tr>
<tr><td>Height</td><td>no data specified</td><td>single data point</td><td>two or more data points</td></tr>
<tr><td>below 10m</td><td>same as at 10m</td><td>same as at 10m, unless explicitly specified</td><td>same as at 10m, unless explicitly specified</td></tr>
<tr><td>10m</td><td>A stability: SIGMA_THETA = 25
B stability: SIGMA_THETA = 20
C stability: SIGMA_THETA = 15
D stability: SIGMA_THETA = 10
E stability: SIGMA_THETA = 5
F stability: SIGMA_THETA = 2.5</td><td>use power law to correct the given sigma theta to a reference height of 10m, if not already given at 10m</td><td rowspan="3">linear interpolation scheme

(Warning to be issued if positive gradient of 5 deg / 100m exceeded)

(Assumes SIGMA_THETA of 1 deg at ML_HGT and above if not explicitly specified and not implied by the slope of the highest two outer points)</td></tr>
<tr><td>0.02 ML_HGT
0.04 ML_HGT
0.07 ML_HGT
0.10 ML_HGT</td><td colspan="2">use power law</td></tr>
<tr><td>0.2 ML_HGT
0.4 ML_HGT
0.7 ML_HGT
1.0 ML_HGT</td><td colspan="2">A-D stability:
Continue with power law up to 0.9 ML_HGT, then taper to 1 deg at 1.1 ML_HGT.
E&F stability:
Linear interpolation between value at 0.1 ML_HGT and 1 deg at ML_HGT</td></tr>
</table>

SIGMA_THETA_HGT

Array (MAXTIM = 100 by MAXLEVEL_ALL = 35) of floating point values specifying the height (m) above terrain of the SIGMA_THETA array values. SIGMA_THETA and SIGMA_THETA_HGT must always be entered as pairs. The interpolation algorithm assumes that heights are ordered from lowest to highest.

III.B.9.g. ADPIC_EXPL_CLD namelist

The ADPIC_EXPL_CLD namelist directs the explosive cloud rise model in ADPIC. However,.whether or not variables are set using this namelist, the explosive cloud rise calculation will only be made if the flag USE_EXPL_CLD_RISE is set in the previous ADPIC.NML namelist ADPIC_CONTROL

DIF_IN_EXPL_CLD

Character ('Y' or 'N') specifying whether or not diffusion is permitted for particles within the explosive cloud.

(Default: DIF_IN_EXPL_CLD = 'Y' → diffusion is permitted within the explosive cloud)

DTDZ1

Floating point value representing the lapse rate from ground to elevation of lowest layer (specified by parameter ZI). Degrees C / m.

DTDZ2

Floating point value representing the lapse rate for elevations higher than the lowest layer ,ZI. Degrees C / m.

HEAMT

Array (max size = number of explosive cloud sources, currently limited to 1) of floating point values representing the mass (# of TNT equivalent) of high explosive contributing to the explosion for each explosive cloud source.

HEATDET

Array (max size = number of explosive cloud sources, currently limited to 1) of floating point values representing the heat of detonation (J/kg) for each explosive cloud source. For high explosive detonation, use HEATDET = 5.52E6 J/kg.

PLOT_EXPL_CLD

Integer specifying the time interval (sec) between ADPIC particle plots during the dynamic explosive cloud rise period of the ADPIC calculation. The possible range is from 1 through 60; if an out-of-range value is specified, the default value is used.

(Default: PLOT_EXPL_CLD = 60 → 60 second interval between particle plots)

PRINT_EXPL_CLD

Flag ('Y'/'N') which controls creation and printing of the output text file EXPL_CLD.PRT.

(Default: 'N')

PRINT_TYPE_EXPL_CLD

5-character string ('SHORT'/'FULL') which controls the amount of output to the output text file EXPL_CLD.PRT.

(Default: 'SHORT')

REL_HUM

Floating point value (0 < REL_HUM < 1.0) representing the relative humidity (fraction) at the surface in the explosive cloud source area.

SIG_OVER_R

Floating point value which is a multiplier which converts the explosive cloud radius into the initial sigma. It is applied in all three directions, i.e., the initial sigma in each sirection is

sig = sig_over_r * r_expl_cld

(Default: SIG_OVER_R = 10.)

SRC_EXPL_CLD

Double-dimensioned array (max size = ADPIC source counter by the number of explosive cloud sources, currently limited to 1) of integers (values = ADPIC source numbers) specifying which ADPIC source (1 to MAXSOURC) goes into which explosive cloud source. The first index is a counter for whichof the MAXSOURC possible ADPIC sources will be used with the specificied explosive cloud source (the second index). For example, “SRC_EXPL_CLD(1,1) = 2 3 6” indicates that ADPIC sources #2, #3, and #6 will be contained in a single explosive cloud source #1. NOTE: MULTIPLE EXPLOSIVE CLOUD MODELING IS NOT YET IMPLEMENTED!!!!!

SRC_PRESS

Floating point value representing the atmospheric pressure (mb) at the height of the initial explosive cloud source.

TA0

Floating point value representing the ground level temperature, degrees C.

TEMP_PROFILE

Array (max size = MAX_EXPL_CLD_PROF = 30), giving an air temperature profile (C) at various levels in the atmosphere, at heights specified by TEMP_PROFILE_HGT).

TEMP_PROFILE_HGT

Array (max size = MAX_EXPL_CLD_PROF = 30), representing the height (m) above ground level for the temperature profile given in the array TEMP_PROFILE.

UAEXP

Floating point value representing the wind power-law exponent.

UAREF

Floating point value representing wind speed (m/s) at the reference height UAZREF.

UAZREF

Floating point value representing the wind speed reference height (m).

VEL_COUPL

Floating point value representing the velocity coupling coefficient. The suggested value of "VEL_COUPL = 0.35" was determined from the Clean Slate I dataset of the Roller Coaster experiment.

ZI

Floating point value representing the height (m) of the lowest layer.

ZINIT

Array (size = number of explosive cloud sources, currently limited to 1) of floating point values representing the initial height (m) above ground level of the center of each explosive cloud source.

Sample ADPIC.NML Namelist Input File

```
$adpic_control                                                    namelist title
 PROBLEM_TITLE     = 'Jul 91: Run1G - Stability & src hgt'        title for plots

 RUN_START_DATE              = '01APR85'                         start of run
 RUN_START_TIME              = '00:00'
 RUN_DURATION                = '00::04:00'                       duration of run
 SAMPLING_INTERVAL           = '00::01:00'                       interval for CONC files
 RESTART_WRITE_INTERVAL      = '00::00:00'                       interval for .RST files
 PLOT_INTERVAL               = '00::01:00'                       dot plot interval
 STATUS_INTERVAL             = '00::00:15'                       screen write interval
 INITIAL_TIME_STEP           = 30                                initial time step
 DIFF_METHOD                 = 'RDM'                             diffusion method

 SOURCES_TO_SAMPLING_BIN(1,1)  = 1
 SOURCES_TO_SAMPLING_BIN(1,2)  = 1
 SOURCES_TO_SAMPLING_BIN(1,3)  = 1
 SOURCES_TO_SAMPLING_BIN(1,4)  = 2
 SOURCES_TO_SAMPLING_BIN(1,5)  = 2                               bin arrangement
 SOURCES_TO_SAMPLING_BIN(1,6)  = 2
 SOURCES_TO_SAMPLING_BIN(1,7)  = 3
 SOURCES_TO_SAMPLING_BIN(1,8)  = 3
 SOURCES_TO_SAMPLING_BIN(1,9)  = 3
 SOURCES_TO_SAMPLING_BIN(1,10) = 1 2 3
 SAMPLING_TYPE =  'INTAIR'   'INSAIR'   'TOTDEP'  'INTAIR'       types of sampling bins
                  'INSAIR'   'TOTDEP'  'INTAIR'   'INSAIR'
                  'TOTDEP'  'INTAIR'
 SAMPLING_HGT =   1.5  1.5  0.0  1.5  1.5  0.0  1.5  1.5  0.0  75.   hgts of sampling bins
REFLECT_AT_MLHGT = 'N'
REFLECT_AT_GRIDTOP =' Y'
PARTPOS = 'P'                                                   for dots on terrain
 $end                                                           end of namelist

 $ADPIC_METPARAMS                                               namelist title
 MET_START_DATE  =     '01APR85'  '01APR85'  '01APR85'  '01APR85'  time for met params
 MET_START_TIME  =     '00:00'    '01:00'    '02:00'    '03:00'
 INV_OBUKHOV_LEN_SL =0.0          -0.06      0.0        0.06
 PWR_LAW_EXPNT_SL =    0.17       0.13       0.17       030
 INV_OBUKHOV_LEN_ML= 0.0          –0.05      0.0        0.04
 ML_HGT          =     500        750        500        250
 TURB_PARAM_VERT= 4 *'KZ_SIMTHRY'
 TURB_PARAM_HORZ= 4 * 'SIGH_SIGTHETA'
 STAB_CLASS      =     4          2          4          6
 SFC_ROUGH_HGT   =     0.10

 $end                                                           end of namelist
```

$adpic_source					*namelist title*
SPECIES	=	'GND'	'ELEVATED'	'COLUMN'	*species names*
DECAY_MODE = 3 * 'RAD_DECAY'					*hybrid particle flag*
SIGX	=	1.0	1.0	4.0	*std deviation in x*
SIGY	=	1.0	1.0	4.0	*std deviation in y*
SIGZ	=	1.0	1.0	40.0	*std deviation in z*
XPOS_CUTOFF	=	2.5	2.5	10.	*x cutoff to right*
XNEG_CUTOFF	=	-2.5	-2.5	-10.	*x cutoff to left*
YPOS_CUTOFF	=	2.5	2.5	10.	*y cutoff to forward*
YNEG_CUTOFF	=	-2.5	-2.5	-10.	*y cutoff to backward*
ZPOS_CUTOFF	=	2.5	2.5	80.	*z cutoff to top*
ZNEG_CUTOFF	=	-0.	-2.5	-120.	*z cutoff to bottom*
XLOC	=	105.	105.	105.	*x coord of src location*
YLOC	=	135.	135.	135.	*y coord of src location*
TOTPART	=	6000	6000	6000	*total # of particles*
HALFLIFE	=	0.0	0.0	0.0	*decay half life*
SFC_DEP_VEL	=	0.01	0.01	0.01	*deposition velocity*
PART_DIAM_MEDIAN	=	0.3	0.3	0.3	*particle median diam.*
PART_DIAM_SGD	=	3.0	3.0	3.0	*particle std geom dev*
PART_DIAM_MIN	=	0.1	0.1	0.1	*particle min. diameter*
PART_DIAM_MAX	=	100.	100.	100.	*particle max. diameter*
PART_DENSITY	=	1000.	1000.	1000.	*particle density*
USE_PLUME_RISE	=	'N'	'N'	'N'	*plume rise flag*
SOURCE_START_DATE	=	'01APR85'	'01APR85'	'01APR85'	*start for src rates*
SOURCE_START_TIME	=	'00:00'	'02:00'	'05:00'	
SOURCE_RATE(1,1)	=	1.0	0.0	0.0	*rates for 1st species*
SOURCE_RATE(1,2)	=	1.0	0.0	0.0	*rates for 2nd species*
SOURCE_RATE(1,3)	=	0.0	1.0	0.0	*rates for 3rd species*
CENTER_HGT(1,1)	=	0.	100.	120.	*hgts for 1st species*
CENTER_HGT(1,2)	=	0.	100.	120.	*hgts for 2nd species*
CENTER_HGT(1,3)	=	0.	100.	120.	*hgts for 3rd species*
$end					*end of namelist*
$adpic_expl_cld					*namelist title*
ZINIT	=				
HE_AMT	=				
HEATDET	=				
REL_HUM	=	-99999.0			
SRC_PRESS	=	-99999.0			
TEMP_PROFILE_HGT	=				
TEMP_PROFILE	=				
$END					*end of namelist*

III.B.10. ADPIC.CLST

PLCNT can optionally read a list of fully qualified, and therefore unique, CONC***.BIF filenames from ADPIC.CLST. This file is produced by ADPIC and contains the list of CONC***.BIF files created by the most recentADPIC run. This is an ASCII file that can be edited by an assessor and provides a mechanism to easily handle multiple CONC groups in a single directory by allowing the user to maintain a series of ADPIC.CLST files (under different names with the desired file being copied to ADPIC.CLST as needed). In the absence of an ADPIC.CLST file, PLCNT processes the entire set of unique CONC***.BIF qualifiers in the appropriate directory, taking the highest version number associated with each qualifier.

III.B.11. CONC_ANALYSIS.NML

CONC_ANALYSIS.NML is the input file that directs the execution of both PLCNT and TIMEHIS. In PLCNT it supports control of the types of plots produced and the times when they are produced, the dose and final bin processing, the contour levels and numerous details of the plot format and labeling. TIMEHIS is supported by the same dose and final bin processing and the same time controls as PLCNT but for the interpolation of point concentration values.

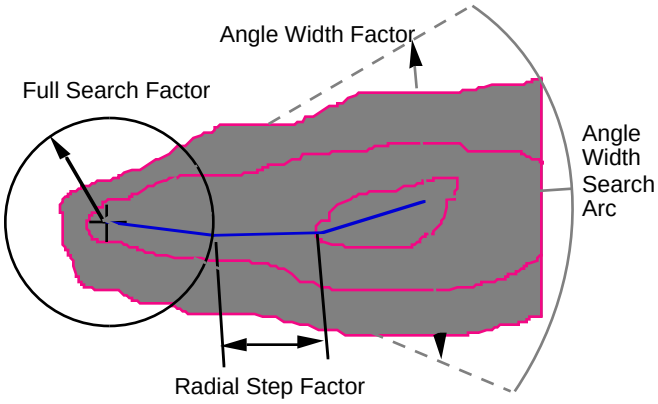
The following is a list of the CONC_ANALYSIS.NML variables grouped alphabetically within namelist blocks: CONC_ANALYSIS_CONTROL, PLCNT_GRAPHICS_PARAMS, PLCNT_LABELING_PARAMS, and TIMEHIS_PARAMS. The first namelist block is used by both codes, the next two are used only by PLCNT, and the last block only by TIMEHIS. This order must be maintained by all namelist blocks present in the file, although unused blocks can be omitted.

III.B.11a. CONC_ANALYSIS_CONTROL namelist

It should be noted that if Hybrid Particle ADPIC is used, the dose conversion factors have already been applied. Therefore the dose bins in CONC_ANALYSIS.nml should not include dose conversion factors.

ANGLE_WIDTH_FACTOR

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the distance tagent to the previous maximum over which the centerline search is performed. A line of the specified length is evaluated perpendicularly to a line from the source to the previous centerline point at that centerline point. The centerline function then calculates an arc tangential to this line and the search for the next centerline value is determined along this arc:



This parameter takes over at radii beyond the FULL_SEARCH_FACTOR. This feature works best with a plume that moves directly away from the source. It can have problems with a meandering plume, especially a plume that ends up oriented radially around the source. To avoid the zig-zags, it is best to leave this value low (e.g., 2 to 4); however if the value is too low the maximum along an arc may be missed.

(Default: ANGLE_WIDTH_FACTOR = 3.0)

CENTERLINE_SOURCE

Integer (1 to MAXSOURC) representing the ADPIC source whose location is to be used as a starting point for the centerline processing. This is the only required input for the centerline capability; if it is missing, then the centerline function is not performed.

(Default: CENTERLINE_SOURCE = missing)

CONC_RESTART_DATE

7-character string (ddmmmyy) specifying the restart date of an ADPIC*.RST file. PLCNT will read the corresponding CONC*.RST file and accumulate concentration data from all CONC*.BIF files with times greater than the restart time and generate contour plots. Note that there must be at

least one CONC*.BIF file with a time greater than the restart time. This feature enables the user to not have to have all of the CONC*.BIF files in a single directory in order to make a PLCNT run which needed integrated air concentration data.

CONC_RESTART_TIME

8-character string (hh:mm:ss) specifying the restart time of an ADPIC*.RST file. PLCNT will read the corresponding CONC*.RST file and accumulate concentration data from all CONC*.BIF files with times greater than the restart time and generate contour plots. Note that there must be at least one CONC*.BIF file with a time greater than the restart time. This feature enables the user to not have to have all of the CONC*.BIF files in a single directory in order to make a PLCNT run which needed integrated air concentration data.

DCON

Array (max size = MAXFBINS = 30) of values, each of which act as a numerical multiplier to the concentrations calculated by ADPIC for a dose bin. The size of the array is required to be equal to the number of dose bins (maximum of 30) specified in SAMPLING_TO_DOSE_BIN. The DCON parameter must either be specified completely or be omitted, in which case the concentrations in the dose bins remain unchanged. The actual units of the bin contents to be plotted by PLCNT are determined by a combination of SOURCE_RATE (an ADPIC.NML parameter), DCON, and the ADPIC model units (s/m^3 for instantaneous and time-averaged, s^2/m^3 for integrated, s/m^2 for deposition):

actual units = (SOURCE_RATE) × (DCON) × (ADPIC model units)

There are several possible combinations involving SOURCE_RATE and DCON; below are four of the most common:

1) SOURCE_RATE = actual source rate (e.g., g/s or Ci/s),
   DCON = dose conversion factor (dcf)
2) SOURCE_RATE = unit source rate in 1/s,
   DCON = dcf and the actual amount (e.g., g or Ci)
3) SOURCE_RATE = unit amount (instead of a rate),
   DCON = dcf × actual source rate (e.g., g/s or Ci/s)
4) SOURCE_RATE = dcf × actual source rate (e.g., g/s or Ci/s),
   DCON = 1

Combination #1 is the one used by the model parameters system. Combination #4 is currently necessary when dose plots for multi-nuclides are desired.

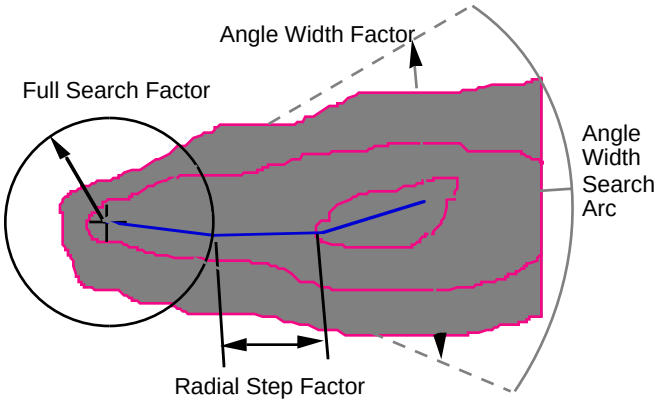
DOSE_TO_FINAL_BIN

2-D array (max size =MAXDBINS, MAXFBINS = 30) of integers (values = dose bin numbers) specifying which dose bin numbers will be combined into final bin *n*, i.e., the DOSE TO FINAL bin correlation. This allows for specification of the combination of dose bins to be delayed until

PLCNT/TIMEHIS, as well as allowing for the re-use of the dose bins. (Also see SAMPLING_TO_DOSE_BIN.)

FULL_SEARCH_FACTOR

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the radial distance over which to perform a search from SECTOR_ANGLE_MIN counterclockwise to SECTOR_ANGLE_MAX (usually 0 to 360 degrees) for a concentration maximum; thus the ANGLE_WIDTH_FACTOR is ignored within this radius:



This search was designed to be the initial search pattern to find out which way the plume is going. It is not recommended to make this value very high, since in a splotchy, well spread out plume it will cause a zig-zag centerline that will follow individual high points across the map. A value of 1 to 3 should be adequate for simple plumes. In a complex plume case where the plume initially extends one direction, then breaks up, and later extends in a different direction, you may need to use a larger FULL_SEARCH_FACTOR to find the beginning of the later extension.

(Default: FULL_SEARCH_FACTOR = 3.0)

INTEGRATION_STEP

A two-dimensional array (max plot series,max final bins) of 13-character strings (ddd::hh:mm:ss) that specifies the time interval over which interval integrations are to be performed plots for a given plot series. The INTEGRATION_STEP must be an integer multiple of the SAMPLING_INTERVAL in ADPIC.NML and less than or equal to the PLOT_STEP for the same plot series. By default the INTEGRATION_STEP for a plot series is equal to the PLOT_STEP.

INTEGRATION_TYPE

An array (max final bins) of four-character strings ('CUMM', 'INTV', or 'NONE') that specifies the type of integration to perform for a given final bin. 'CUMM' refers to cumulative integration, 'INTV' refers to interval integration, 'NONE' refers to no integration. Only 'CUMM' or 'INTV' can be used for an integrated sampling bin (i.e., SAMPLING_TYPE='INTAIR' in the adpic_control namelist).

NEST_GRIDS

Flag ('Y' or 'N') which determines whether the nested grid or the coarse sampling grid is to be used.

(Default: NEST_GRID = 'Y' → used nested grid.)

PLOT_END_DATE

A two-dimensional array (max plot series, max final bins) of 7-character strings (ddmmmyy) that specifies the date portion of the date/time when a given plot series for a given final bin is to end. The date/times for each final bin should be increasing order. Specifying an end date/time for a plot series is optional. If it is omitted, then a given plot series is terminated by the beginning of a second series or by the of the available concentration data. A plot end date/time must corresponding to the date/time of a CONC*.BIF file.

PLOT_END_TIME

A two-dimensional array (max plot series, max final bins) of 8-character strings (hh:mm:ss) that specifies the time portion of the date/time when a given plot series for a given final bin is to end. The date/times for each final bin should be increasing order. Specifying an end date/time for a plot series is optional. If it is omitted, then a given plot series is terminated by the beginning of a second series or by the of the available concentration data. A plot end date/time must corresponding to the date/time of a CONC***.BIF file.

PLOT_START_DATE

A two-dimensional array (max plot series, max final bins) of 7-character strings (ddmmmyy) that specifies the date portion of the date/time when a given plot series for a given final bin is to begin. The date/times for each final bin should be increasing order. A plot start date/time must corresponding to the date/time of a CONC***.BIF file.

PLOT_START_TIME

A two-dimensional array (max plot series, max final bins) of 8-character strings (hh:mm:ss) that specifies the time portion of the date/time when a given plot series for a given final bin is to begin. The date/times for each final bin should be increasing order. A plot start date/time must corresponding to the date/time of a CONC***.BIF file.

PLOT_STEP

A two-dimensional array (max plot series, max final bins) of 13-character strings (ddd::hh:mm:ss) that specifies the time interval between plots for a given plot series. The PLOT_STEP must be an integer multiple of the SAMPLING_INTERVAL in ADPIC.NML.

PRINT_CONTOUR_CONC

Flag ('Y', 'N') for writing final bin concentrations to the echo file. This writing also includes the location and value of the max/min concentrations.

(Default: PRINT_CONTOUR_CONC = 'N'→ none written.)

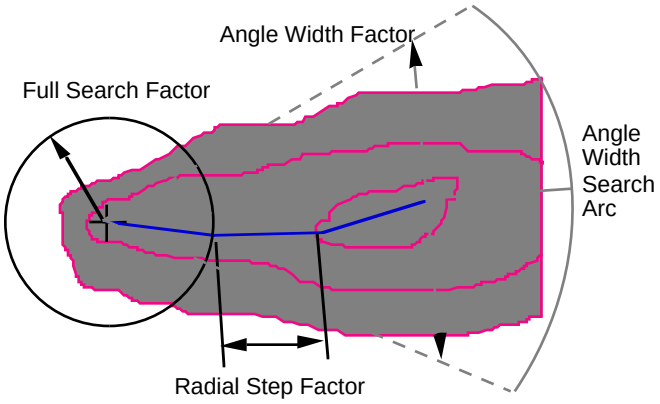
PRINT_INPUT_CONC

Flag ('Y', 'N') which controls the printing of the input concentrations to the echo file.

(Default: PRINT_INPUT_CONC = 'N'→ none are written.)

RADIAL_STEP_FACTOR

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the radial step distance for the search for each maximum.



(Default: RADIAL_STEP_FACTOR = 0.5)

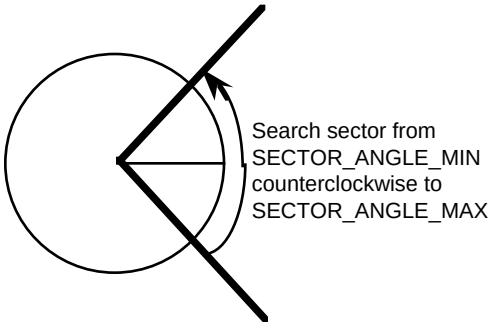
SAMPLING_TO_DOSE_BIN

Array (max size = MAXFBINS = 30) of integers, whose values are ADPIC bin numbers, specifying which sample bin goes into which dose bin, i.e., the "ADPIC TO DOSE" bin mapping. For example, if the same sample bin were put into three different dose bins, different dose conversion factors (DCON's) could be applied to each dose bin (for example: whole body dose, child thyroid dose, and air concentration) without having to have separate sample bins. (Also see DOSE_TO_FINAL_BIN.)

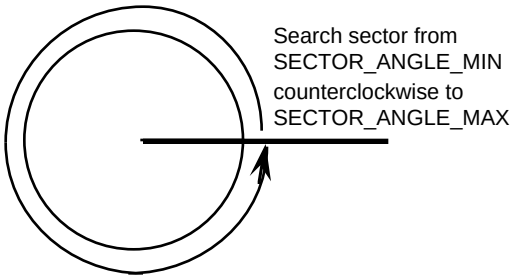
(Default: SAMPLING_TO_DOSE_BIN = 1 2 3 4 5 6 7 8 9 10 → one dose bin for each sample bin from ADPIC, with the dose bins in the same order as the sample bins)

SECTOR_ANGLE_MAX

Floating point value (0. to 360.) limiting all centerline searches (close-in full search, and further-out angle width search). SECTOR_ANGLE_MAX is measured in degrees counterclockwise from east (a mathematical angle, not a meteorological angle). Centerline searches cover angles counterclockwise from SECTOR_ANGLE_MIN to SECTOR_ANGLE_MAX:



Example with SECTOR_ANGLE_MIN = 315 deg
and SECTOR_ANGLE_MAX = 45 deg

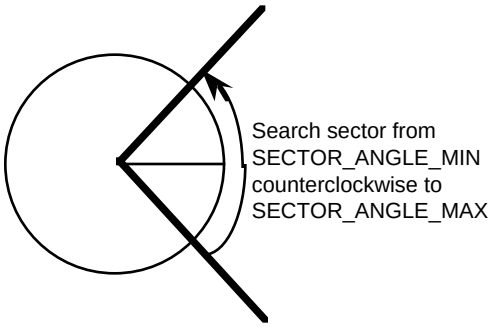


Example with SECTOR_ANGLE_MIN = 0 deg
and SECTOR_ANGLE_MAX = 360 deg

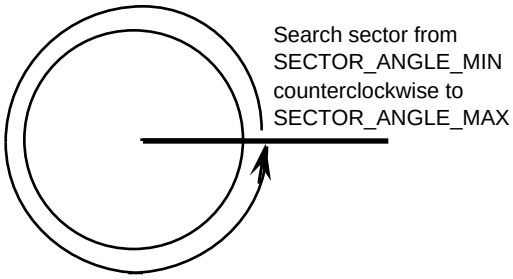
(Default: SECTOR_ANGLE_MAX = 360.)

SECTOR_ANGLE_MIN

Floating point value (0. to 360.) limiting all centerline searches (close-in full search, and further-out angle width search). SECTOR_ANGLE_MIN is measured in degrees counterclockwise from east (a mathematical angle, not a meteorological angle). Centerline searches cover angles counterclockwise from SECTOR_ANGLE_MIN to SECTOR_ANGLE_MAX:



Example with SECTOR_ANGLE_MIN = 315 deg
and SECTOR_ANGLE_MAX = 45 deg



Example with SECTOR_ANGLE_MIN = 0 deg
and SECTOR_ANGLE_MAX = 360 deg

(Default: SECTOR_ANGLE_MIN = 0.)

III.B.11b. PLCNT_GRAPHICS_PARAMS namelist

BLOWUP_SOURCE

An integer specifying the number of the ADPIC source whose location is to be placed as indicated by SOURCE_OCTANT.

CIRCLE_2000

Flag ('Y' or 'N') specifying whether or not the 2000ft blast effects circle will be plotted at the release location. (CIRCLE_2000 = 'Y' is required for DOD plots.)

CONTOUR_LEVELS

A two-dimensional array (MAXCONTLEVELS=5, MAXFBINS=30) of specific contour levels to be plotted for each specified final bin. Thus, up to 5 contour levels can be specified for each final bin.

The CONTOUR_LEVELS parameter is required for DOD and AWE plots as follows (for final bin n):

CONTOUR_LEVELS(1,n)	=	150.	25.	5.	0.5	(DOD CEWBDE)
CONTOUR_LEVELS(1,n)	=	600.	60.	6.		(DOD depostition)
CONTOUR_LEVELS(1,n)	=	4.7	0.94	0.78	0.15	(AWE 50yr CEWBDE)
CONTOUR_LEVELS(1,n)	=	20.0	2.0	0.2	0.03	(AWE depostition)

If CONTOUR_LEVELS(1,n) is not specified, PLCNT will pick its own contour levels, based upon the MIN_CONTOUR_AREA requirement. Also see RECOMPUTE_USER_CONTOURS for note regarding interaction among MIN_CONTOUR_AREA, CONTOUR_LEVELS, and RECOMPUTE_USER_CONTOURS.

HALF_POWER_OF_10

Flag indicating whether half powers of 10 (i.e., $3*10^{n)}$ are to be allowed as possible contour levels.

(Default: HALF_POWER_OF_10 = 'Y')

INITIAL_BANNER

Flag ('Y', 'N') used to indicate an initial calculation. The fact that this is an Initial calculation will be part of the plot description record. (Note: the sending of the description and graphics files is now controlled interactively in Show_Graphics-Site_Workstation_Plot and Show_Graphics-Site_System_Plot.)

(Default: INITIAL_BANNER = 'N' → not an Initial calculation)

MAPSCALE

A two-element integer array specifying the map scale to be used. The first element represents a unit of actual distance on the map and the second element is the corresponding actual physical

distance. For example, a map scale of 1:50,000 means and 1 inch on the map represents 50,000 inches of actual physical distance, and MAPSCALE = 1 50000. This parameter is ignored if MAP_SCALE_FLAG = 'N' for all final bins.

MAP_SCALE_FLAG

Array (max size = MAXFBINS = 30) of flags ('Y', 'N') that indicates whether or not the specified map scale is to be applied to this bin.

(Default: MAP_SCALE_FLAG = 'N')

MAX_CONTOURS

Array (max size = MAXFBINS = 30) of integers (1,2,3,4, or 5) of contour levels to be plotted and shown in legend.

MIN_CONTOUR_AREA

Real number representing minimum area (km^2) that will be plotted and shown in the legend. This will eliminate any contour levels specified by CONTOUR_LEVELS which do not meet the MIN_CONTOUR_AREA requirement. (Also see RECOMPUTE_USER_CONTOURS for note regarding interaction among MIN_CONTOUR_AREA, CONTOUR_LEVELS, and RECOMPUTE_USER_CONTOURS.)

(Default = 0.001% of grid area or $0.01km^2$, whichever is larger)

NEST_TYPE

10-character string which specifies if and how the nested grids are shown on the contour plots. Current values and meanings are:

'NONE' → no nesting stuff drawn
'BOUNDARIES' → boundaries of nested grids drawn
'CELLS' → grid cells and boundaries of nested grids drawn

NUM_RECOMPUTED_CONTOURS

Integer indicating the number of contours to recompute if they are recomputed (See RECOMPUTE_USER_CONTOURS).

(Default: NUM_RECOMPUTED_CONTOURS = 5)

OUTPUT_GRIDS

Flag ('Y', 'N') indicating whether PLCNT is to produce products to be sent to the site workstation. When products send to the site_workstation are the final bin concentrations passed as grids of data.

(Default: OUTPUT_GRIDS = 'Y')

PLOT_SIZE_CM

A real number specifying the size (cm) of the resulting plot as it would appear on the hard copy device. The postscript printers used by ARAC use a value of 16.0 for PLOT_SIZE_CM.

RECOMPUTE_USER_CONTOURS

Flag ('Y', 'N') indicating whether PLCNT should generate contours levels based on the maximum concentration if the estimated area if the lowest user-specified contour level is less than the MIN_CONTOUR_AREA (See NUM_RECOMPUTED_CONTOURS). See the table below for the interaction between CONTOUR_LEVELS, MIN_CONTOUR_AREA, and RECOMPUTE_USER_CONTOURS in determining isopleths to be plotted. Feedback to the user will be provided on the screen during model runtime if any of the alternative isopleth selection methods are attempted. (Also see NUM_RECOMPUTED_CONTOURS)

Choice	Isopleth selection method	MIN_CONTOUR_AREA
1	use values from CONTOUR_LEVELS	use this
2	recompute if RECOMPUTE_USER_CONTOURS='Y'	use this
3	use values from CONTOUR_LEVELS	ignore this
4	recompute if RECOMPUTE_USER_CONTOURS='Y'	ignore this

(Default: RECOMPUTE_USER_CONTOURS = 'Y')

SMOOTHED_CONC_SWS

Flag ('Y', 'N') indicating whether or not smoothing is to be performed along nest boundaries for site workstation products. Graphics artifacts due to the nesting can occasionally occur with the contouring algorithm and this option may permit more presentable products to be generated. Smoothing is also reccommended for gridded concentrations to be used with ARCINFO.

(Default: SMOOTHED_CONC_SWS = 'N' => no smoothing will be done)

SOURCE_OCTANT

2-character string ('NW','N ','NE','W ','C ','E ','SW','S ','SE') specifying the desired placement of the source within the first subplot when scaling maps (MAP_SCALE_FLAG = 'Y'). The following diagram indicates the two character codes corresponding to the allowed locations:

NW	N	NE
W	C	E
SW	S	SE

XMAX_WINDOW

Real number specifying the maximum *x* coordinate of the area to be windowed. Used in conjunction with XMIN_WINDOW, YMIN_WINDOW, and YMAX_WINDOW.

(Default: none, i.e., no windowing.)

XMIN_WINDOW

Real number specifying the minimum *x* coordinate of the area to windowed. Used in conjunction with XMAX_WINDOW, YMIN_WINDOW, and YMAX_WINDOW.

(Default: none, i.e., no windowing.)

YMAX_WINDOW

Real number specifying the maximum *y* coordinate of the area to be windowed. Used in conjunction with XMIN_WINDOW, XMAX_WINDOW, and YMIN_WINDOW.

(Default: none, i.e., no windowing.)

YMIN_WINDOW

Real number specifying the minimum *y* coordinate of the area to be windowed. Used in conjunction with XMIN_WINDOW, XMAX_WINDOW, and YMAX_WINDOW.

(Default: none, i.e., no windowing.)

III.B.11*c*. PLCNT_LABELING_PARAMS namelist

DESCR_FORMAT

Array (max size = MAXFBINS = 30) of 5-character strings specifying the format to be used in displaying the cross-hatching and area information in the plot legend. Current values and meanings are: '3LINE' → use 3-line DOD action level format, '2LINE' → use 2-line format, 'CODED' → use 2-character coded format which does not include the isopleth value (used for AWE), and 'NONE' → no coding or action level descriptions.

DESCR_HEADERA

Array (max size = MAXFBINS = 30) of 38-character strings specifying the first line of a two-line header that appears immediately above the cross-hatching section of the legend. (Usual values: DESCR_HEADERA = 'Exposure Action Levels:' for DOD sites and DESCR_HEADERA = 'Contours:' for other sites.

DESCR_HEADERB

Array (max size = MAXFBINS = 30) of 38-character strings specifying the second line of a two-line header that appears immediately above the cross-hatching section of the legend. (Usual value: DESCR_HEADERB = '(Level and Area Covered) ' .

GENERATION_DATE

7-character string (ddmmmyy) labeled as the generation date of a set of plots in the second or third line of the plot legend. Note that this time is not currently the same as the generation time in the product description record.

(Default: GENERATION_DATE = 'unset' which uses the current system time.)

GENERATION_TIME

8-character string (hh:mm:ss) labeled as the generation time of a set of plots in the second or third line of the plot legend. Note that this time is not currently the same as the generation time in the product description record.

(Default: GENERATION_TIME ='unset' which uses the current system time.)

LEGEND_TITLE

Array (max size = MAXFBINS = 30) of 38-character strings, which appear in the margin above the legend section.

(Default: LEGEND_TITLE =' '.)

LEGEND_TITLE_ALL

32-character string for all final bins, which appears in the margin above the legend section.

(Default: LEGEND_TITLE_ALL =' '.)

LEVEL_DESCRit

Array (max size = MAXFBINS = 30) of 38-character strings describing emergency or protective actions associated with a given contour level (specified by CONTOUR_LEVELS). There are a maximum of three (t = A, B and C) 38-character strings each for each of 5 contour levels (i = 1 to 5) for each of 30 final bins. Up to 3 lines of 38 characters may be used for each of a maximum of 4 contour levels for DESCR_FORMAT = '3LINE', up to 5 lines of 38 characters may be used for each of the maximum of 5 contour levels for DESCR_FORMAT = '2LINE', and up to 8 characters of the 1st line (LEVEL_DESCRiA string) may be used for each of 4 contour levels for DESCR_FORMAT = 'CODED'. Note that LEVEL_DESCRit is not used for DESCR_FORMAT (n) = 'NONE'. Below is a complete specification (i.e., explicitly specifying all three lines for each of the four possible contour levels) for DESCR_FORMAT = '3LINE' for final bin 1:

```
LEVEL_DESCR1A (1)    =  Immediate action required:
LEVEL_DESCR1B (1)    =  Respiratory protection required.
LEVEL_DESCR1C (1)    =  Recommend sheltering or removal.

LEVEL_DESCR2A (1)    =  Prompt respiratory protection
LEVEL_DESCR2B (1)    =  required and recommend
LEVEL_DESCR2C (1)    =  evacuation.

LEVEL_DESCR3A (1)    =  Respiratory protection required,
LEVEL_DESCR3B (1)    =  recommend sheltering,
LEVEL_DESCR3C (1)    =  consider evacuation.

LEVEL_DESCR4A (1)    =  Consider sheltering.
LEVEL_DESCR4B (1)    =
LEVEL_DESCR4C (1)    =
```

MATERIAL

Array (max size = MAXFBINS = 30) of 22-character strings specifying the material or species for a specified final bin that is to appear in the plot legend.

PLOT_TITLE

Array (max size = MAXFBINS = 30) of 38-character strings which will appear in the margin above the plot. This string will also appear across the top of the graphics section.

(Defaults: PLOT_TITLE = 'CONTOUR PLOT')

REMARKSt

Array (max size = MAXFBINS = 30) of 38-character strings for up to 6 lines (t = A to F) of remarks.

REMARKSt_ALL

38-character strings for up to 6 lines (t = A to F) of remarks that are not final bin-specific.

SOURCE_LOCATION_NUM

Integer (0 to number of sources) representing the ADPIC source number for which the location of the source should be displayed in the plot legend. For example, “SOURCE_LOCATION_NUM = 1” will result in location of ADPIC source number 1 being written in legend.

(Default: SOURCE_LOCATION_NUM = 0 → no location is written in the legend.)

UNITS

Array (max size = MAXFBINS = 30) of 10-character strings specifying the units of the concentrations of the specified final bin that appear on the plots.

NOTE: When DESCR_FORMAT = '3 LINE,' only the first 6 characters of UNITS will appear on the plots.

(Default: UNITS = ' '.)

III.B.11.d. TIMEHIS_PARAMS namelist

BACKGROUND_CALC

Array (max size = MAXFBINS = 30) of floating point values representing the background concentration to be added to the model-calculated concentration values before statistics are calculated. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

BACKGROUND_CALC_ALL

Floating point value for all final bins representing the background concentration to be added to the model-calculated concentration values before statistics are calculated.

BACKGROUND_MEAS

Array (max size = MAXFBINS = 30) of floating point values representing the background concentration to be added to the measured concentration values (input via the SAMINFO parameter) before statistics are calculated. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

BACKGROUND_MEAS_ALL

Floating point value for all final bins representing the background concentration to be added to the measured concentration values (input via the SAMINFO parameter) before statistics are calculated.

BIN_PLOT_MAX

Array (max size = MAXFBINS = 30) of floating point values that sets the maximum on the vertical axis for the time series plots. It is final bin-specific, so a different maximum can be used for all samplers associated with a given final bin.

BIN_PLOT_MAX_ALL

Floating point value which sets the maximum on the vertical axis for the time series plots for all final bins.

BIN_PLOT_MIN

Array (max size = MAXFBINS = 30) of floating point values that sets the minimum on the vertical axis for the time series plots. It is final bin-specific, so a different minimum can be used for all samplers associated with a given final bin.

BIN_PLOT_MIN_ALL

Floating point value which sets the minimum on the vertical axis for the time series plots for all final bins.

COMPARE_MEAS_CALC

Flag ('Y', 'N') controlling whether statistics between measured and calculated values are calculated. If this flag is set in 'Y', measured-calculated comparison statistics including percentage of values within factor “R” will be written to the STATS.DAT file, and the minimum and maximum computed concentrations for various hit angles will be written to the SAMPOUT.DAT file. The SAMPOUT.DAT file is used as input for another statistics code, FN. The COMPARE_MEAS_CALC flag also controls plotting of measured data on time history graphics plots.

FACTOR_R

Array (max size = MAXFACTORS = 15) of floating point values representing factors to be used in factor R analysis. Current values are (1.5, 2.0, 3.0, 5.0, 7.0, 10.0, 20.0, 50.0, 100.0, 1000.0).

HIT_ANGLE

Array (max size = MAXHANGLES = 10) of floating point values representing angles to be used in hit angle analysis. These angles are used to define an area centered on an interpolation point where sample of additional interpolations are calculated. If a measured value lies within the range defined by the minimum and maximum concentration found in the sample the a hit (or exact match) is recorded for this angle (See Section II.C.4 for a description of this the hit angle area). Current values are (0.0, 2.0 5.0, 10.0, 15.0, 20.0). Note that the first hit angle is always zero and is used to record the interpolated value for the specified location.

MIN_CONC

Array (max size = MAXFBINS = 30) of floating point value representing the minimum measured or computed concentration to be used for statistics calculations. If either the measured or computed concentrations (with background values already added in) are greater than MIN_CONC, then that measured-computed pair is included in the statistics. The default is 0.0. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

MIN_CONC_ALL

Floating point value for all bins representing the minimum measured or computed concentration to be used for statistics calculations. If either the measured or computed concentrations (with background values already added in) are greater than MIN_CONC, then that measured-computed pair is included in the statistics. The default is 0.0.

PROBLEM_TITLE _TIMEHIS

70-character string that is written as the first line of the TIMEHIS.LOG file.

SHOW_TIME_ORIGIN

Flag ('Y', 'N') used to ensure that the time origin (i.e, 0 hours) is included on time history plots. Used to produced plots that are appropriate for publication. Default is 'N'.

STATISTICS_SOURCE

Integer indicating the source to be used in hit angle analysis. That is, the angles and radii used are with reference to a particular ADPIC source, so this source must be specified. Default is 1.

TIMESTEPS_PER_PLOT

Integer number (maximum of 200) of data points (corresponding to SAMPLING_INTERVAL restart files) to be plotted on each frame of the time history plots.

XMAX_SAMPLES

Real number specifying the maximum *x* coordinate of the area to be analyzed. Used in conjunction with XMIN_SAMPLES, YMIN_SAMPLES, and YMAX_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

XMIN_SAMPLES

Real number specifying the minimum *x* coordinate of the area to analyzed. Used in conjunction with XMAX_SAMPLES, YMIN_SAMPLES, and YMAX_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

YMAX_SAMPLES

Real number specifying the maximum *y* coordinate of the area to be analyzed. Used in conjunction with XMIN_SAMPLES, XMAX_SAMPLES, and YMIN_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

YMIN_SAMPLES

Real number specifying the minimum *y* coordinate of the area to be analyzed. Used in conjunction with XMIN_SAMPLES, XMAX_SAMPLES, and YMAX_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

Sample PLCNT Input File

```
$conc_analysis_control
SAMPLING_TO_DOSE_BIN   =  1 2 3 4 5 6 7 8 9 10          ←  relation of ADPIC bins to dose bins
DCON          =  1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0  ←  dose conv. factors for each dose bin
DOSE_TO_FINAL_BIN(1,1)  =  1     4     7
DOSE_TO_FINAL_BIN(1,2)  =  2     5     8                  }  relation of dose bins to final bins
DOSE_TO_FINAL_BIN(1,3)  =  3     6     9
DOSE_TO_FINAL_BIN(1,4)  =  10
PLOT_START_DATE(1,1)    =  '01APR85'
PLOT_START_DATE(1,2)    =  '01APR85'                     }  date (ddmmmyy) to start a plot
PLOT_START_DATE(1,3)    =  '01APR85'                        series
PLOT_START_DATE(1,4)    =  '01APR85'
PLOT_START_TIME(1,1)    =  '01:00'
PLOT_START_TIME(1,2)    =  '01:00'                       }  time (hh:mm) to start a plot series
PLOT_START_TIME(1,3)    =  '01:00'
PLOT_START_TIME(1,4)    =  '01:00'
PLOT_STEP(1,1)          =  '00::01:00'
PLOT_STEP(1,2)          =  '00::01:00'                   }  time interval (dd::hh:mm) between
PLOT_STEP(1,3)          =  '00::01:00'                      plots
PLOT_STEP(1,4)          =  '00::01:00'
INTEGRATION_TYPE(1)     =  'cumm'
INTEGRATION_TYPE(2)     =  'none'                        }  specify type of time integration
INTEGRATION_TYPE(3)     =  'none'
INTEGRATION_TYPE(4)     =  'none'
NEST_GRIDS              =  'Y'                           ←  flag for use of nested grid sampling
$end

$plcnt_graphics_params
INITIAL_BANNER          =  'N"                           ←  Initial plot flag for expanded descr.
CIRCLE_2000             =  'N'                           ←  flag for 2000ft circle
MIN_CONTOUR_AREA        =  0.0                           ←  minimum contour area
NEST_TYPE               =  'NONE          '              ←  nesting grid to be overlayed on plot
MAX_CONTOURS(1)         =  5
MAX_CONTOURS(2)         =  5                             }  maximum number of isopleths (for
MAX_CONTOURS(3)         =  5                                each final bin)
MAX_CONTOURS(4)         =  5
CONTOUR_LEVELS(1,1)     =  1E0   1E-1   1E-2   1E-3   1E-4
CONTOUR_LEVELS(1,2)     =  1E-4  1E-6   1E-8   1E-10                }  isopleths to be plotted
CONTOUR_LEVELS(1,3)     =  1E-1  1E-2   1E-3   1E-4   1E-5          (for each final bin)
CONTOUR_LEVELS(1,4)     =  1E0   1E-1   1E-2   1E-3   1E-4
$end

$plcnt_labeling_params
PROBLEM_TITLE_PLCNT     =  'ARAC Contour Plots'          ←  title for expanded desc.
MATERIAL(1)             =  'Gnd, Elevated, Column '
MATERIAL(2)             =  'Gnd, Elevated, Column '      }  material labels for plot
MATERIAL(3)             =  'Gnd, Elevated, Column '         legend (for each final bin)
MATERIAL(4)             =  'Gnd, Elevated, Column '
```

```
UNITS(1)                =  'Ci*s/m3'
UNITS(2)                =  'Ci/m3'
UNITS(3)                =  'Ci/m2'
UNITS(4)                =  'Ci*s/m3'
GENERATION_DATE         =  'unset'
GENERATION_TIME         =  'unset'
SOURCE_LOCATION_NUM     =  1
PLOT_TITLE(1)       =  'Jan 1991 Benchmark Runs'
PLOT_TITLE(2)       =  'Jan 1991 Benchmark Runs'
PLOT_TITLE(3)       =  'Jan 1991 Benchmark Runs'
PLOT_TITLE(4)       =  'Jan 1991 Benchmark Runs'
LEGEND_TITLE(1)     =  'Run1G: Src hgts, Stabilities '
LEGEND_TITLE(2)     =  'Run1G: Src hgts, Stabilities '
LEGEND_TITLE(3)     =  'Run1G: Src hgts, Stabilities '
LEGEND_TITLE(4)     =  Run1G: Src hgts, Stabilities '
REMARKSA_ALL        =  Remarks: STAB_CLASS=4, 2, 4, 5 '
REMARKSB_ALL        =  'Winds: calm,lgt, med, stg '
REMARKSC_ALL        =  5m diam. gnd & elevated spheres, '
REMARKSD_ALL        =  200m column sources '
DESCR_HEADERA(1)    =  'Exposure Action Levels:  '
DESCR_HEADERB(1)    =  '(Level and Area Covered) '
DESCR_HEADERA(2)    =  'Exposure Action Levels:  '
DESCR_HEADERB(2)    =  '(Level and Area Covered) '
DESCR_HEADERA(3)    =  'Exposure Action Levels: '
DESCR_HEADERB(3)    =  '(Level and Area Covered) '
DESCR_HEADERA(4)    =  Exposure Action Levels: '
DESCR_HEADERB(4)    =  '(Level and Area Covered) '
DESCR_FORMAT(1)     =  'NONE '
DESCR_FORMAT(2)     =  'NONE '
DESCR_FORMAT(3)     =  'NONE '
DESCR_FORMAT(4)     =  'NONE '
$end
```

units labels for plot legend (for each final bin)

← plot generation date & time for plot legend

← src number for nesting

2nd line of plot legend (for each final bin)

1st line of plot legend (for each final bin)

4 lines of free-form remarks for plot legend (optionally final bin-specific)

specifies content of 2 lines forming the subtitle for the isopleth labeling section of the plot legend

specifies format of isopleth labeling in plot legend (for each final bin)

$timehis_params

```
PROBLEM_TITLE_TIMEHIS  =  'ARAC Time History Plots'
TIMESTEPS_PER_PLOT     =  8
COMPARE_MEAS_CALC      =  'Y'
BACKGROUND_CALC_ALL    =  0.000E+00
BACKGROUND_MEAS_ALL    =  0.000E+00
MIN_CONC_ALL           =  0.000E+00
$end
```

← title for plots

← timesteps/points per plot

← compare meas. to calc. values

← backgnd values to be added

← min. value for statistics calc.

III.B.12. TIMEHIS

The CONC_ANALYSIS.NML file used to control PLCNT is also used to control TIMEHIS. TIMEHIS uses the CONC_ANALYSIS_CONTROL and the TIMEHIS_PARAMS namelist blocks in this file, which are described in sections III.B.11.a and III.B.11.d, respectively. Sampler data is entered from a separate file, described next.

The SAMPLER.DATA file has one required section and one optional section. The required section contains the sampler names, coordinates and associated final bins at which concentration values are to be interpolated from the model concentration data. If measurement data is available, this is included in an optional section. Each entry in this section contains the sampler name, the associated final bin number, the observation data, the observation time and the value. Note that if the sampler name and final bin fields are replaced with commas the value from the previous record is inherited. Thus, redundantly specifying these values is unnecessary. Also note that the keyword SAMPLERS is required (unquoted, on a separate line) in all SAMPLER.DATA files and the MEASUREMENTS keyword is required (also unquoted, on a separate line) if measurements data is to be included.

Example SAMPLER.DATA file

```
SAMPLERS
'S0001'     500.0    4201.1    2   3   4          sampler name, x-coord,
'S0002'     503.5    4215.0    4                  y-coord, array (max size=15)
                                                  of final bins
'S0003'     495.11   4191.2    2   3              (Maximum of 400 sampler
'S0004'     513.8    4203.9    2   4              locations)

MEASUREMENTS
'S0002'     2        '11JAN93'  '1200'    121.0   sampler name, final bin,
     ,      ,        '11JAN93'  '1300'    105.0   meas. date, meas. time,
     ,      ,        '11JAN93'  '1400'    95.0    measurement value
     ,      ,        '11JAN93'  '1500'    78.0    (Maximum of 200
     ,      ,        '11JAN93'  '1600'    51.0    measurement times)
     ,      3        '11JAN93'  '1200'    149.0
     ,      ,        '11JAN93'  '1300'    137.0
     ,      ,        '11JAN93'  '1400'    97.0
'S0003'     4        '11JAN93'  '1300'    110.0
     ,      ,        '11JAN93'  '1400'    93.0
     ,      ,        '11JAN93'  '1500'    70.0
     ,      ,        '11JAN93'  '1700'    56.0
     ,      ,        '11JAN93'  '1900'    45.0
     ,      ,        '11JAN93'  '2100'    35.0
     ,      ,        '11JAN93'  2200'     159.0
```

III.C. Output Files

III.C.1. TOPOG*.GRID

The GRID file is generated by TOPOG and supplies the model grid definitions and model terrain for all subsequent codes in the ARAC code stream. MEDIC, MATHEW, ADPIC, PLCNT and TIMEHIS read the GRID file. GRID is an ASCII file used to pass data between codes and provides the coordinate information for the entire model sequence. For any meaningful use of the binary files that used in other parts of the ARAC model system, the TBB pointer to the appropriate GRID file must be maintained as the final description of the grid exists only in this file. The GRID file is composed of two sections:

Variable	Dimension	Type	Description
Traceback Block			Traceback file names
MODELTOPO	IMAXMAT,JMAXMAT	INTEGER	Model topography ; MATHEW grid 0 origin vertical cell index

MODEL_TOPO

The model topography is represented by an integer array with each array element (i,j) corresponding to a horizontal MATHEW grid cell. The value of each array element as the number of *k* levels above the grid bottom, i.e., the bottom of the grid indicated by zero (0): this is the ADPIC convention for vertical cell indexing. The array is dimensioned IMAXMAT by JMAXMAT which, for the standard R41 MATHEW grid, is 51 by 51. With this data, MEDIC and MATHEW must increment the cell heights by one after reading this array and ADPIC must selection the appropriate subset of the array as a function of the ADPIC grid dimensions relative to the MATHEW grid dimensions.

III.C.2. TOPOG.LOG

TOPOG is the ASCII output file for TOPOG. It is used to examine and verify the results of, or to debug, a TOPOG run. It is not used by other computer codes. During normal execution of the ARAC code stream, this file would rarely be examined, but in the event of difficulty, the information is available. There are four sections in an TOPOG file:

1. TOPOG.NML echo
2. Grid parameter display
3. Cell heights
4. Model levels

Each of these sections is discussed below.

Cell Heights

The averaged cell height in meters for each MATHEW cell is printed. This is stored in a real array dimensioned IMAXMAT by JMAXMAT.

Grid Parameter Display

The various grid parameters are written exactly as they are displayed on the user's terminal during a TOPOG run. These parameters include the minimum and maximum extent of the terrain data base in UTM coordinates, the MATHEW and ADPIC model grid dimensions, the grid cell sizes, and the sampling cell description. If errors are found in defining the model grids, then the error messages are printed before the parameter display as they are on the user's terminal.

Model Levels

The MATHEW model level (k-levels) for each cell is printed. This is stored in an integer array dimensioned IMAXMAT by JMAXMAT.

III.C.3. TOPOG*.CNML

The entire TOPOG.NML namelist block is written to a uniquely namedexactly as it appears in the input file if the TOPOG.NML data are read correctly. Otherwise the last line will contain the error.

III.C.4. TOPOG*.CGMB

TOPOG*.CGMB is the graphics output file for TOPOG. It is composed of two plots:

1. Topography contours,
2. MATHEW cell heights. These are described as follows:

MATHEW Cell Heights

The MATHEW cell height along each column of grid cells is the number of model levels below the terrain with one (1) indicating the lowest terrain level in the MATHEW grid.

TOPOGRAPHY Contours

The topography contours are isolines of model terrain height. The MATHEW block topography is converted to heights in meters above sea level at each of the horizontal grid points. The height at a model grid point is equal to the highest of the grid cells (usually four) adjacent to it. The resulting array of values is contoured and the contour lines are labeled in meters above sea level. The default contour interval in 1*DELZ but this can be overridden by giving a value to CONTOUR_INTERVAL

in TOPOG.NML (see TOPOG.NML description for details). The contours are displayed on the MATHEW grid; the ADPIC grid is indicated by a square drawn within the grid. To avoid confusion with the contour lines, no line elements of geography are drawn, but the point elements (location names, character data) are displayed.

III.C.5. MEDVEL***.BIF

The MEDVEL files are generated by MEDIC and are used to pass the extrpolated or interpolated wind field to MATHEW. One MEDVEL file is produced for each data set specified by a MEDIC_PARAMS namelist. The MEDVEL is a binary file composed of two sections described below:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
U0, V0	IMAXMAT,JMAXMAT,KMAXMAT	REAL*4	Extrapolated winds

Traceback Block

See Appendix C

U0, V0

The extrapolated or interpolated winds from MEDIC or HMEDIC are written in separate *u*- and *v*-component arrays as floating point numbers in meters/second. Wind vectors exist at each grid point in the MATHEW grid volume. Therefore, each array is dimensioned IMAXMAT by JMAXMAT by KMAXMAT. For the standard R41 MATHEW grid the arrays are 51 by 51 by 15. The grid volume includes terrain. The component arrays follow each other (*v* after *u*) in the MEDVEL file.

III.C.6. MEDIC.LOG

MEDIC.LOG is the ASCII output file from MEDIC. One MEDIC.LOG file is produced for each MEDIC run containing This file is used to examine and verify the results of, or to debug, a MEDIC run. It is not used by other computer codes. During normal execution of the ARAC code stream, this file would rarely be examined, but if problems arise the information is available as an aid in locating the difficulty.

There are five possible sections in each MEDIC.LOG file which are discussed belwo:

1. MEDIC.NML listing
2. Grid descriptions
3. Tables of valid GWC data if input
4. Reference level winds (MEDIC) or GWC winds (HMEDIC)
5. Extrapolated or interpolated MEDIC winds

MEDIC.NML listing

The completed MEDIC.NML data set is written out including default settings. The MEDIC.NML listing will appear in all MEDIC.LOG files.

Grid Descriptions

The model grid parameters are listed under a 'TOPOG.NML:' heading. This includes values describing the terrain data base, ADPIC, and MATHEW grids.

Tables of valid data input from GWC metdata files

Tables are printed indicating the valid input data read from each AFGWC metdata file accessed.

Reference Level or GWC Winds

The reference level wind vectors (MEDIC) or the input GWC winds (HMEDIC) are printed as separate *u*- and *v*-component arrays of real numbers in meters second. A reference level wind vector exists at each grid point in the horizontal direction, so the arrays are dimensioned IMAXMAT by JMAXMAT. The *u*-component array is printed first, followed by the *v*-component array. This section will appear only if PRINT_WINDFIELDS is set to 'Y' in the STNLOC.MET file.

Extrapolated or InterpolatedWinds

The extrapolated or interpolated wind vectors are printed as separate *u*- and *v*-component arrays of real numbers in meters/second. These arrays are printed as a series of two-dimensional arrays, one for each vertical model level. Thus for a typical R41 grid, 15 arrays, each dimensioned 51 by 51, will be printed to define one component. The *u*-component arrays are written first, followed by the *v*-component arrays. This section will appear only if PRINT_WINDFIELDS is set to one 'Y' in the STNLOC.MET file.

IIII.C.7. MEDIC*.CGMB

MEDIC*.CGMB is the graphics output file for MEDIC. It is composed of seven types of plots:

1. Surface data wind barb plots (MEDIC only)
2. Upper air wind barb plots (MEDIC and HMEDIC)
3. Horizontal plane vector plots (MEDIC and HMEDIC)
4. Sample profile locations (MEDIC and HMEDIC)
5. Sample profiles (MEDIC and HMEDIC)
6. Measured profiles (MEDIC only)
7. Input data (MEDIC only)

These plots are described below:

Surface Data

The surface wind observations,are plotted on the MATHEW grid border as wind barbs. For regional MEDIC, the winds are adjusted to the reference height (REF_HGT) according to the surface layer power law using the exponent PWRSL.

Wind barbs, as used on ARAC graphics products, present a slight variation on the standard meteorological wind barb convention for representing wind observations. In the standard convention, wind direction is represented by a line segment, or stem, with one end, the origin, located at the observation point and the other end located exactly upwind from the origin. Thus the stem represents the direction from which the wind is blowing. Speed ranges are represented by the number of flags and short line segments, i.e., barbs, attached to the stem. The first flag or barb is attached to the stem at the distal end with additional flags and barbs being placed closer to the origin. Flags are added first, at the distal end, followed by barbs. Each flag represents a wind speed of 50 knots and each barb represents 10 knots. The last barb, i.e., closest to the origin, may be half the length of a normal barb, thereby indicting a 5 knot wind. Thus a stem with one flag, one barb, and one half barb indicates a wind of at least 50 + 10 + 5 = 65 knots, but less than 70 knots (see Fig. B.1a.). With ARAC's convention, because many of the surface wind speeds are low, the 5 knot (~ 2.5 m/s) resolution of this scheme is limiting. Therefore the last barb is scaled exactly to the reported speed, e.g., a 12 knot wind would be represented as one barb at the end of the stem (10 knots) and one short barb one-fifth as long as a standard barb somewhat closer to the origin (see Fig. III.C.1b.).

Horizontal Plane Vector Plots

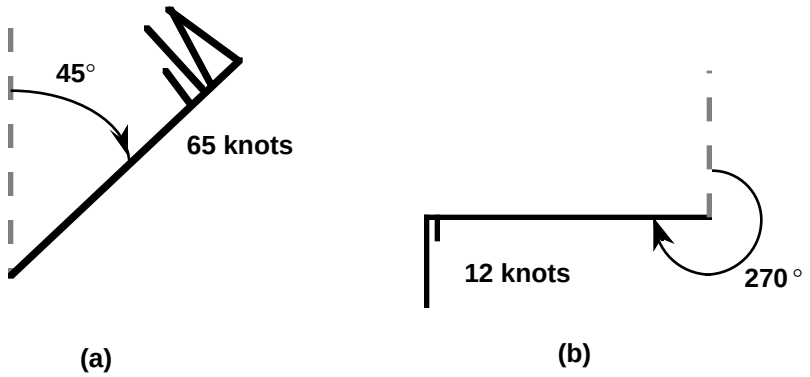


Fig. III.C.1. Examples of wind barbs showing (a) 65 knot winds from 45 N and (b) 12 knot winds from 270 N.

In plotting the surface observations, the station name is printed adjacent to the origin of the barb. If the speed data is missing from an observation, only the stem is plotted, indicating the direction. If the direction data is missing from an observation, a square is plotted, containing an integer giving the wind speed in m/s. A zero wind speed observation is indicated by a double circle. The density of the barbs can be controlled by assigning value to the appropriate variables

GWC_BARB_PLOT_FREQ, RWM_BARB_PLOT_FREQ, or SFC_BARB_PLOT_FREQ. Similarly, the vectors may be scaled using GWC_BARB_SCALE, RWM_BARB_SCALE, or SFC_BARB_SCALE.

Horizontal grids of extrapolated wind vectors are plotted at constant heights above terrain. Since wind vectors crossing the outer boundary of the MATHEW grid are clipped, vectors are plotted only at internal grid points (1<I<MAT and 1<J<JMAXMAT). This offers some protection against clipping. The density of the vectors can be controlled by giving a value to VECTOR_PLOT_FREQ in MEDIC. The default is two (2), which causes vectors to be plotted at every other grid point (see MEDIC description for details). The size of the vectors can be scaled by giving values to the parameter HORZ_VECTOR_SCALE in MEDIC (see MEDIC description for details). The levels plotted by default are the reference level, 1*DELZ, 6*DELZ, and the upper level winds. All these levels are at constant heights above terrain. The upper level winds are the winds at the vertical grid extent (i.e.KMAX*DELZ) above terrain, unless the parameterized extrapolation is being used and the top of the boundary layer (BL_HGT) is greater than KMAX*DELZ. In this case the upper level winds are the winds at the top of the boundary layer. Other model level wind vectors can be plotted by assigning values to PLOT_LEVELS in MEDIC (see MEDIC description for details).

Sample Profile Locations

The positions of the sample profiles are indicated by asterisks (*) within the MATHEW grid outline. The asterisks are followed by numbers used to identify the profiles.

Sample Profiles

The extrapolated winds along a column of MATHEW grid points are displayed as a function of height. Three different profiles are plotted with the vertical extent of each representing the vertical dimension of the grid, (KMAX*DELZ), above the local terrain. The first display is a profile of speed as a function of height and the second is a profile of direction as a function of height; both profiles are plotted as solid lines. The last display is a vertical line with wind barbs attached. The origins of the wind barbs are placed at various model levels and indicate the extrapolated winds at that level. (See the discussion on “surface data” for a description of wind barbs.) Every other *k*-level is shown, as well as the wind at the reference height (REF_HGT), the top of the surface layer (SL_HGT), and the top of the boundary layer (BL_HGT). The speeds and directions at the model grid levels are printed below the displays. If the profile extrapolation procedure is being used, the speed and direction profiles synthesized from the upper air data are shown on the corresponding profile displays as dotted lines. These profiles were used to generate the extrapolated winds along this column of grid points and are helpful in understanding the effects of some of the extrapolation parameters. (See the MEDIC model description for details of the extrapolation process.) If default values are used, plots will be generated for nine columns of grid points spaced evenly over the MATHEW grid area. The user may select different columns by assigning values to IPROFILE and JPROFILE or XPROFILE and YPROFILE in MEDIC (see MEDIC description for details).

Measured Profiles

Speed and direction profiles and wind barbs are plotted for upper air station observations below the top of the model grid. Asterisks on the profile plots and barbs are shown only at levels for which there is data If all of the speed or direction data for a station is missing, then the corresponding profile does not appear.

Input Data

All the input data used in TOPOG and MEDIC is listed where appropriate from the files:

1. TOPOG.NML
2. MEDIC.NML
3. OBSERV.MET
3. STNLOC.MET

III.C.8. MATVEL

MATVEL files are generated by MATHEW and contain the advection (mass-balanced) winds used by ADPIC. Each MATVEL file contains a single data set (i.e., adjusted wind fields created from a single MEDVEL data set). The MATVEL files are binary files used to pass data between codes and is not normally examined by a user. MATVEL file data sets are composed of three data sections:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
UPIC,VPIC,WPIC	IMAXPIC,JMAXPIC, KMAXPIC	REAL*4	ADPIC advection winds

These sections are described as follows:

Traceback Block

See Appendix C for a description of the traceback block.

UPIC, VPIC, WPIC

The advection winds are a subset of the MATHEW grid point adjusted winds chosen to cover the ADPIC grid. These winds are stored in three arrays, one for the *u*-components, one for the *v*-components, and one for the *w*-components of the wind vectors which are expressed as real floating point numbers in m/s. These wind vectors exist at each ADPIC grid point, so each array is dimensioned IMAXPIC by JMAXPIC by KMAXPIC which, for a typical R41 ADPIC grid, is 41 by 41 by 15. These component arrays are stored one after the other (*u*, *v*, *w*) in the MATVEL file data set.

III.C.9. MATHEW.VLST

MATHEW writes a list of fully qualified, and therefore unique, MATVEL*.BIF filenames to MATHEW.VLST. This file contains the list of MATVEL***.BIF files created by a given MATHEW run. It is read by ADPIC as an optional way of controling which MATVEL***.BIF files to process in a given run. This is an ASCII file that can be edited by a user and provides a mechanism to easily handle multiple MATVEL groups in a single directory by allowing the user to maintain a series of MATHEW.VLST files (under different names with the desired file being copied to MATHEW.VLST as needed).

III.C.10. MATHEW.LOG

MAT is the ASCII output file for MATHEW. One MAT file is produced for each dataset number, i.e., for each new value of NRUN. This file is used to examine and verify the results of, or to debug, a MATHEWrun. It is not used by other computer codes. During normal execution of the ARAC codestream, this file would rarely be examined, but if problems arise, the information is available as an aid in locating the difficulty. There are as many as nine sections in an MAT file:

1. MATHEW echo
2. Problem data description
3. Grid point extrapolated winds
4. Face-centered extrapolated winds
5. Extrapolated wind divergence
6. Lagrangian multipliers
7. Face-centered adjusted winds
8. Grid point adjusted winds
9. ADPIC adjusted winds

Each of the sections is discussed below:

ADPIC Adjusted Winds

The ADPIC wind vectors are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the *w*-components. Also each array is dimensioned IMAXPIC by JMAXPIC by KMAXPIC (41 by 41 by 15 for a typical R41 ADPIC grid). This section will appear only if PRINT_ADJUSTWIND_ADPIC is set to 'Y' in the MATHEW.NML file.

ADPIC Adjusted Wind Divergence

The ADPIC adjusted wind divergences are printed in the same format as the grid point extrapolated winds. This section will appear only if PRINT_ADJUSTWIND_DIVERG is set to 'Y' in the MATHEW.NML file.

Extrapolated Wind Divergence

The extrapolated wind divergences are printed in the same format as the grid point extrapolated winds. This section will appear only if PRINT_INITIALWIND_DIVERG is set to 'Y' in the MATHEW.NML file.

Face-Centered Extrapolated Winds

The face-centered extrapolated wind components are printed in the same format as the grid point extrapolated winds (see 3 above). This section will appear only if PRINT_INITIALWIND_FACE is set to 'Y' in the MATHEW.NML file.

Face-Centered Adjusted Wind

The face-centered adjusted wind components are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the*W*-components after writing the other two arrays. This section will appear only if PRINT_ADJUSTWIND_ADJU is set to 'Y' in the MATHEW.NML file.

Grid Point Extrapolated Winds

The grid point extrapolated winds are printed as *u*- and *v*-component arrays of real numbers in m/s. These arrays are printed as a series of two-dimensional arrays, one for each vertical model level. Thus for a typical R41 ARAC grid, 15 arrays, each dimensioned 51 by 51, will be written to define one component. The *u*-component arrays are written first, followed by the *v*-component arrays. This section will appear only if PRINT_INITIALWIND_GRID is set to 'Y' in the MATHEW.NML file.

Grid Point Adjusted Winds

The grid point adjusted wind vectors are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the *w*-components after writing the other two arrays. This section will appear only if PRINT_ADJUSTWIND_GRID is set to 'Y' in the MATHEW.NML file.

Lagrangian Multipliers

The Lagrangian multipliers are printed, after convergence, in the same format as the extrapolated wind divergence. This section will appear only if PRINT_LAMBDAS is set to 'Y' in the MATHEW.NML file.

MATHEW Echo

The MATHEW data set is written exactly as it appears in the input file if the MATHEW data was read correctly. Otherwise the line containing the error will be the last to appear.

Problem Data Description

All parameters and data affecting the problem solution to this point are printed under four headings which indicate the source of the parameters. The subsections are:

'TOPOG.NML' -which includes data defined by the terrain data as well as by TOPOG.NML

'MATHEW' -which lists the MATHEW parameters, including any default values.

III.C.11. MATHEW*.CGMB

MATHEW*.CGMB is the graphics output file for MATHEW. As a default, it is composed of four types of plots:

1. Horizontal plane vector plots
2. Vertical profile locations
3. Vertical wind profiles
4. Input data

These plots are described as follows:

Horizontal Plane Vector Plots

Horizontal grids of adjusted MATHEW wind vectors are plotted as constant heights above terrain. These vectors are plotted within the MATHEW grid borders. Since wind vectors crossing the outer boundary of the MATHEW grid are clipped, vectors are plotted only at internal grid points (1<I<IMAX and 1<J<JMAX). The density of the vectors can be controlled by setting a value to VECTOR_PLOT_FREQ in MATHEW. The default is two (2), which causes vectors to be plotted at every other grid point (see MATHEW description for details). By default only the MATHEW winds 1*DELZ above terrain are shown. By selecting model levels through PLOT_LEVELS and MATHEW, the user can view as many or as few model levels as desired.

Vertical Profile Locations

The position of the vertical profiles are indicated by asterisks (* within the MATHEW grid boundary. The asterisks are followed by numbers used to identify the profiles.

Vertical Wind Profiles

The adjusted wind vectors along a column of MATHEW grid points are displayed as a function of height. Three different displays are plotted, each with the vertical extent representing the vertical range of the MATHEW grid so terrain is included in the plots. The first display is a profile of speed as a function of height and the second is a profile of direction as a function of height. The last display

is a vertical line representing the vertical extent of the grid with wind barbs attached at every other model *k* level. These barbs represent the adjusted wind at the various levels (see Section 1 of GRAFMEDIC for a discussion of wind barbs). The wind speeds and directions at the model *k* levels are printed below the displays. If the default value is used, plots are generated for nine columns of grid points spaced evenly over the MATHEW grid area. The user may select different columns by assigning values to IPR and JPR in MATHEW (see MATHEW description for details).

Input Data

All the input data used by TOPOG, MEDIC, and MATHEW is listed where appropriate under four headings:

1. TOPOG.NML
2. MATHEW

III.C.12. CONC*.BIF

CONC files are generated by ADPIC and contain deposition and instantaneous and integrated air concentration information used by PLCNT in plotting the contours. A CONC FILE is written at each output time and contains five two-dimensional arrays of concentration for each level, up to MAXSBINS levels. Each CONC*.BIF is a binary file which is rarely examined by the user. It consists of the following data:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
NUM_BINS	1	Integer*4	Number of sample bins
NUM_NESTS	1	Integer*4	Number of nested grids
X_ADPIC_GRID	IMAXPIC	REAL*4	Model x coords of CELL_CONC array
Y_ADPIC_GRID	JMAXPIC	REAL*4	Model y coords of CELL_CONC array
X_NESTED_GRID	I_MAX_NEST MAX_NEST_DEPTH	REAL*4	Model x coords of CELL_CONC_NESTED arrays
Y_NESTED_GRID	J_MAX_NEST MAX_NEST_DEPTH	REAL*4	Model y coords of CELL_CONC_NESTED arrays
CELL_CONC	IMAXPIC JMAXPIC NUMBERBIN	REAL*4	Concentration arrays
CELL_CONC_NESTED	IMAXPIC JMAXPIC NUMBERBIN MAX_NEST_DEPTH	REAL*4	Nested grid concentration arrays

This variable information is described as follows:

The Traceback Block (TBB) information includes first and last vel file names and portions of the vel file TBB from MEDIC and MATHEW.

CELL_CONC

These are a series of two-dimensional concentration arrays, where the type of concentration, species sampled, and sampling height have been specified by the user in ADPIC. Up to MAXSBINS) combinations of SAMPLING_TYPE, SOURCES_TO_SAMPLING_BIN, and SAMPLING_HGT can be specified in any given run.

CELL_CONC_NESTED

These are a series of four two-dimensional nested grid arrays of concentration that correspond to the CELL_CONC arrays (see Section II.B.3.e. for a detailed explanation).

NUM_BINS

This variable specifys the number of sample bins

NUM_NESTS

This is the number of nested grids used by ADPIC.

X_ADPIC_GRID

X_ADPIC_GRID values are the Model x coordinates of the sampling grid cell centers whose concentrations are given in the CELL_CONC array.

Y_ADPIC_GRID

Y_ADPIC_GRID values are the Model y coordinates of the sampling grid cell centers whose concentrations are given in the CELL_CONC array. When the elements in Y_ADPIC_GRID are matched with those in X_ADPIC_GRID, the spatial coordinates of the cell centers are completely specified.

X_NESTED_GRID

X_NESTED_GRID values are the Model x coordinates of the sampling grid cell centers whose concentrations are given in the CELL_CONC_NESTED array.

Y_NESTED_GRID

Y-NESTED_GRID values are the Model y coordinates of the sampling grid cell centers whose concentrations are given in the CELL_CONC_NESTED array. When the elements in Y_NESTED_GRID are matched with those in X_NESTED_GRID, the spatial coordinates of the cell centers are completely specified.

III.C.13. RECEPTOR.DAT

The RECEPTOR.DAT file is generated by ADPIC whenever the moving receptors option (USE_MOVING_RECEPTORS = 'Y' in ADPIC_CONTROL namelist) is used. (See Section III.B.9.e

ADPIC_RECEPTOR namelist for more information on the moving receptors option.) The RECEPTOR.DAT file contains the following information:

- A user-specifiable title line
- The ADPIC run start time
- For each specified receptor:
  1) The receptor number and receptor ID
  2) A table with the following columns of information:
     a) Elapsed run time in seconds
     b) Elapsed run time in ddd::hh:mm:ss format
     c) Integrated air concentration (cumulative integration)
     d) Averaged air concentration for the current time interval

If no concentrations were observed for a given receptor, a statement to that effect is written to the RECEPTOR.DAT file.

III.C.14. ADPIC.LOG

ADPIC.LOG is a ASCII output file from ADPIC. This file is not used by other codes and it is rarely examined by the user. However, it can serve as a diagnostic aid by helping the user locate the source of a fatal error, when such an error occurs during an ADPIC run. There can be up to six sections in an ADPIC.LOG file:

1. ADPIC.NML Echo
2. New Source Rates
3. Traceback Block
4. Particle Budget
5 Particle Summary
6. Optional Output

Each section is discussed below.

ADPIC.NML Echo

The ADPIC.NML namelist data is written exactly as it appears in the input file, with default values being applied to all values NOT input in the ADPIC.NML namelist file. This information is also contained in the file ADPIC*.CNML, part of the traceback system.

New Source Rates

TIme-varying source rates are incorporated into an ADPIC calculation by using the input array SOURCE_RATE in ADPIC. A line will be added to the ADPIC.LOG file whenever the source rate is changed; this gives the value of the new rate and the time it is valid.

Traceback Block

The Traceback Block contains all the data from the TOPOG.NML, OBSERV.MET, STNLOC.MET, MATHEW, and ADPIC.NML input files. Additionally, a section with the heading MAX VELOCITIES WITH LOCATION INDICES is provided. This section gives the maximum interpolated *u*- and *v*-component winds with their *i*, *j* and *k* indices and the maximum adjusted *u*-, *v*- and *w*-components with their indices. If the particles appear to be moving too fast, a check in this section should indicate whether the problem is caused by advection. The next entries are entitled ITER LIMIT and ITER ACTUAL; they define the maximum number of iterations that are allowed for a MATHEW convergence and the actual number of iterations required to adjust the winds for mass consistency.

Particle Budget

If DIFF_METH = 'RDM', at the end of each global time step, ADPIC write the following information to ADPIC.LOG file:

NCYC - Number of the current global time step cycle
DELT - Global time step(hours:minutes:seconds)
TIME -Total elapsed time (days::hours:minutes:seconds)
since RUN_START_DATE,RUN_START_TIME
TOTLP ACTIVE - Total number of active particles
NUM_GEN PER CYC - Number of particles generated this cycle
TOTLP GEN - Number of particles generated for entire problem
LCYC - Number of particles lost this cycle
TOTLP LOST - Total number of particles lost for entire problem
AVERAGE LOCAL DT - Average value of the particle local time steps during this global time step.

If DIFF_METH = 'GRADI', at the end of each cycle (Δt), ADPIC writes the following information to the ADPIC.LOG file:

NCYC - Number of the current cycle
DELT - Time step for this cycle
TIME - Total time since RUN_START_DATE,RUN_START_TIME
NTOT - Total number of active particles
MGEN - Number of particles generated this cycle
MTOT - Number of particles generated for entire problem
LCYC - Number of particles lost this cycle
LTOT - Total number of particles lost for entire problem
NOCL - Number of times particles moved more than one cell in any direction
AVOCL - Average number of cells that NOCL particles moved

If NOCL becomes significantly larger than one, the diffusion velocity algorithm leads to instabilities. The time step can become erratic under these conditions and the particle plots may show a very uneven distribution, especially in the vertical. If this happens, FRACT should be reduced, resulting in a smaller time step, and the problem rerun. FRACT is input through ADPIC.NML; it has a default value of one (1.).

Particle Summary

At the end of each edit (STATUS_INTERVAL) interval, a partial summary of the particle budget for each source is written to ADPIC. The elements in this list are NTOT, MGEN, and LTOT, and either the particle mass (QSNC) or activity (TACT).

Optional Output

If optional parameters have been selected in ADPIC, they are written to ADPIC.LOG at the end of each edit interval.

III.C.15. ADPIC*.RST

ADPIC*.RST is a binary file which contains all of the information necessary to restart the ADPIC code. The output times (and the permissible restart times) are controlled by the parameter RESTART_WRITE_INTERVAL for uniformly spaced ADPIC*.RST files and with RESTART_WRITE_DATES/RESTART_WRITE_TIMES arrays for arbitrarily spaced ADPIC*.RST FILES. These three parameters/arrays can be set in the Adpic.nml file. More specifically, RESTART_WRITE_INTERVAL indicates the output interval in hours and minutes ('hhmmss') relative to RUN_START_DATE and RUN_START_TIME on the first restart, and RESTART_DATE RESTART_TIME on subsequent restart. If any values are entered in the RESTART_WRITE_DATES/RESTART_WRITE_TIMES arrays, those values override the RESTART_WRITE_INTERVAL. There is currently a limit of ten restart files per ADPIC run.

III.C.16. CONC*.RST

ADPIC accummulates air concentration from the beginning of a run for bins whose sample types are "INTAIR" and dumps this information to CONC*.RST files each time a restart file (ADPIC*.RST) is created. The output format of a CONC*.RST file is identical to the CONC*.BIF file format, except it contains CELL_CONC_CUM and CELL_CONC_CUMC_NESTED data. CONC*.RST files are read by ADPIC on restart runs and by PLCNT to generate contour plots.

III.C.17. PARTPOS*.BIF

Particle position files are generated by ADPIC and can be used directly by graphics post-processing codes to display particle positions or "dots" over terrain images. A PARTPOS file is written at each PLOT_INTERVAL time interval and contains the following data:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
format identifier integer	1	Integer*4	1 for particle format, 2 for video
TOTLP_ACTV	1	Integer*4	current total particle count
X_ORIGIN_PIC	1	Real*4	x coord of lower left grid corner
Y_ORIGIN_PIC	1	Real*4	y coord of lower left grid corner
IMAXPIC	1	Integer*4	x grid size by coordinate
JMAXPIC	1	Integer*4	y grid size by coordinate
KMAXPIC	1	Integer*4	z grid size by coordinate
DELX * 1000.	1	Real*4	x grid spacing in meters
DELY * 1000.	1	Real*4	y grid spacing in meters
DELZ	1	Real*4	z grid spacing in meters
HORIGIN	1	REAL*4	grid base elevation in meters
P_X	TOTLP_ACTV	Real*4	x coordinates of particles in meters
P_Y	TOTLP_ACTV	Real*4	y coordinates of particles in meters
P_Z	TOTLP_ACTV	Real*4	z coordinates of particles in meters
VEL_FILE_AGE	1	Real*4	age (hrs) of last velocity data file used
P_SOURCE_NO	TOTLP_ACTV	Integer*4	originating source # for each particle
P_IDENT	TOTLP_ACTV	Integer*4	unique particle identifier number

III.C.18. ADPIC*.CGMB

ADPIC*.CGMB provides three different views of the Lagrangian particle positions as computed by ADPIC at each PLOT_INTERVAL time. The first plot represents a full projection of the particles onto an x, z plane, with the plane situated over the Model y coordinate of the source point (the observer looks from south to north). The second plot represents a projection of the particles onto a y, z plane positioned over the Model x coordinate of the source (the observer looks from east to west). Topography is transparent in these displays, but individual boundary cells are frequently outlined, in part, when there are numerous particles in the grid volume. The two plots appear on the same page. Since the aspect ratio of the horizontal and vertical grid dimensions is quite large, the two scales in the vertical have been exaggerated to better view the particle distributions.

III.C.19. PLCNT.LOG

PLCNT.LOG is a ASCII output file from PLCNT. This file is not used by other codes, and it is rarely examined by the user. However, if PLCNT fails to run because of an improper input, then PLCNT.LOG can be helpful in locating the source of the fatal error. There can be up to two sections in the PLCNT.LOG file:

1. PLCNT Echo
2. Concentrations

Each section is described below:

Concentrations

The two-dimensional arrays of grid point concentrations which are used by PLCNT in providing the contour plots are written to PLOT. Each array is preceded by all of the descriptor information that appears on the corresponding plot. (See Section III.C.14. for further details.)

PLCNT Echo

These data are written exactly as they appear in the PLCNT file, provided PLCNT has been read correctly. If not, the last line will contain the error.

III.C.20. CENLINE*.DAT

CENLINE*.DAT is a ASCII output file generated from a centerline run of PLCNT. The primary contents of this file are tables of maximum concentrations and their locations (distance in km and bearing from source location, as well as the x and y coordinates)

III.C.21. TIMEHIS.LOG

TIMEHIS.LOG is a ASCII output file from TIMEHIS. This file is not used by other codes, and it is rarely examined by the user. However, if TIMEHIS fails to run because of an improper input, then TIMEHIS.LOG can be helpful in locating the source of the fatal error. There can be up to four sections in the TIMEHIS.LOG file:

1. TIMEHIS Echo
2. Concentrations
3. Samplers

Each section is described below:

Concentrations

The two-dimensional arrays of grid point concentrations which are used by TIMEHIS in providing the contour plots are written to TIMEHIS. Each array is preceded by all of the descriptor information that appears on the corresponding plot. (See Section III.C.14. for further details.)

TIMEHIS Echo

These data are written exactly as they appear in the TIMEHIS file, provided TIMEHIS has been read correctly. If not, the last line will contain the error.

Samplers

If the user decides to place samplers within the ADPIC grid, then the sampler names and their values are printed after the corresponding concentration array. After writing the final set of concentrations to TIMEHIS, TIMEHIS will then output a list of the sampler values.

III.C.22. STATS.DAT

The STATS.DAT file is generated by TIMEHIS, and contains a tabular listing of some statistics divided into four separate sections:

1) Sampler Time Series
   For each final bin specified for each sampler:
   - Sampler ID
   - Final bin number
   - x & y coordinates
   - species or material
   - sampling type
   - sampling height
   - a table listing the following information for each specified output time:
     a) concentration date
     b) concentration time
     c) measured (i.e., sampler) concentration value (-1 indicates no measured value)
     d) computed (from the model) conc. value
     e) ratio of measured to computed concentration values
     f) hit angle (see HIT_ANGLE in TIMEHIS_PARAMS namelist)

2) Matched Measurement Ratios
   For each final bin specified for each sampler:

- Sampler ID
- Final bin number
- x & y coordinates
- a table listing the following information for each output time with non-zero non-missing measured and computed concentration values:
    a) measured (i.e., sampler) concentration value
    b) computed (from the model) conc. value
    c) ratio of measured to computed concentration values
    d) factor (see FACTOR_R in TIMEHIS_PARAMS namelist)
    e) residual
    f) absolute residual

3) Summary Statistics

- A table listing:
    a) bin number
    b) number of observations (measurements)
    c) average measured concentration value
    d) variance of the measured concentration values
    e) average computed concentration value
    f) variance of the computed concentration values
    g) bias
    h) absolute bias
    i) normalized mean square error
- A summary for all matched measurements listing:
    a) average measured-to-computed ratio
    b) standard deviation of the measured-to-computed ratios
    c) number of observations

4) Factor R Frequency Tables

- For each final bin, a table showing how many ratios fall within given ranges of factors
- For all bins, a table showing how many ratios fall within given ranges of factors

Appendix A. Bibliography

Boughton, B.A., and J.M. Delaurentis, 1987: An integral model of plume rise from high explosive detonations. 1987 ASME/AIChE National Heat Transfer Conference, August 9–12, Pittsburgh, PA.

Boughton, B.A., J.M. Delaurentis, and W.E. Dunn 1987: A stochastic model of particle dispersion in the atmosphere, *Boundary-Layer Meteor*., **40**, 147-163.

Briggs, G.A., 1972: Plume rise predictions. Lectures on Air Pollution and Environmental Impact Analyses (Ed. Haugen), *American Meteorological Society*, pp. 53–111, Boston MA.

Businger, J.A., J.C. Wingaard, Y. Izumi and E.F. Bradley, 1971: Flux profile relationships in the atmospheric surface layer. *J. Atmos. Sci*., **28**, 181.

Courant, R. and D. Hilbert, 1962: *Methods of Mathematical Physics*, Interscience Publishers, John Wiley and Sons, New York.

Crandall, W.K., C.R. Molenkamp, A.L. Williams, M.M. Fulk, R. Lange, and J.B. Knox, 1973: An investigation of scavenging of radioactivity from nuclear debris clouds: Research in Progress. Lawrence Livermore National Laboratory Report UCRL-51328.

Crawford, T.V., 1966: A computer program for calculating the atmospheric dispersion of large clouds. Lawrence Livermore National Laboratory Report UCRL-50179.

Davenport, A.G., 1965: The relationship of wind structure to wind loading. National Physical Laboratory Symposium No. 16, *Wind Effects on Buildings and Structures*, pp. 54–102, Her Majesty's Stationery Office, London.

Desiato, F. and R. Lange, 1986: Introduction of plume rise capability into the transport and diffusion codes of the ARIES/ARAC emergency response systems. Lawrence Livermore National Laboratory Report UCID-20652.

Draxler, R.R., 1976: Determination of atmospheric diffusion parameters. *Atmos. Environ*., **10**, 99–105.

Edwards, L. L., and T, F. Harvey, 1984, ARAC-CV2: A Proposed Improvement for ARAC Computational Capability, UASG 84-16.

Environmental Protection Agency (EPA), 1981, “Preface” in *Summary Report of the Regional Workshops on Air Quality Modeling*, EPA, Washington, D.C.

Ermak, D.L. (1992) Dense-Gas Dispersion Advection-Diffusion Model. *Proceedings JANNAF Safety and Environmental Protection Subcommittee meeting*, Aug. 10-14, 1992, Chemical Propulsion Information Agency, CPIA Pub. 588, pp. 215-224. Also, Lawrence Livermore National Laboratory, Livermore, CA, publication UCRL-JC-109697.

Ermak, D.L., J.S. Nasstrom and A.G. Taylor (1995) *Implementation of a Random Displacement Method (RDM) in the ADPIC model framework* , UCRL-ID-121742,

Foster, K.T. and M.H. Dickerson, 1990: An updated summary of MATHEW/ADPIC Model Evaluation Studies. Lawrence Livermore National Laboratory Report UCRL-JC-104134.

Foster, K.T., R. Freis, J.S. Nasstrom, 1990: Incorporation of an explosive cloud rise code into ARAC's ADPIC transport and diffusion model. Lawrence Livermore National Laboratory Report UCRL-ID-103443.

Freis, R. P., T. F. Harvey, and L. L. Edwards, 1989, GENHAZ1, A Program to Calculate Time Dependent Toxicity's, UASG 89-52.

Freis, R. P.,1989 An Analytic Method of Solving the Radioactive Chain Decay Rate Equations, UASG 89-50.

Golder, D., 1972: Relationships among stability parameters in the surface layer. *Boundary Layer Meteorology*, **3**, pp.47–58.

Gresho, P., 1975: *UASG*, 75–22.

Hanna S.R., G.A. Briggs, and R.P. Hosker, Jr. (1982) *Handbook on atmospheric diffusion,* DOE/TIC-11223, U.S. Dept. of Energy, 102 pp..

Harvey, T. F., R. P. Freis, and L. L. Edwards, 1989, Implementation of a Hybrid Particle Version of ADPIC: Phase 1, UASG 89-29.

Kocher, D. 1980, Data set DLC-80/DRALIST, Radiation Shielding Information Center, Oak Ridge National Lab.

Lange, R., 1973: A three-dimensional computer code for the study of pollutant dispersal and deposition under complex conditions. Lawrence Livermore National Laboratory Report UCRL-51462.

Lange, R., 1978: ADPIC—A Three-Dimensional Particle-in-Cell Model for the Dispersal of Atmospheric Pollutants and Its Comparison to Regional Tracer Studies. *J. Appl. Meteoro.*, **17**, 320–329.

Lange, R., 1989: Transferability of a three-dimensional air quality model between two different sites in complex terrain. *J. of Appl. Meteoro.*, **28**, 665–679. Also Lawrence Livermore National Laboratory Report UCRL-96220.

McDonald, J.E., 1960: An aid to computation of terminal fall velocities of spheres. *J. Meteor*., August 1960, p. 463.

McMahon, T.A. and P.J. Denison, 1979P: Empirical atmospheric deposition parameters—A survey. *Atmospheric Environment*, **13**, pp. 571–585.

Nasstrom, J.S. (1995) *Turbulence parameterizations for the RDM version of ADPIC*, UCRL-ID-120965, Lawrence Livemore National Lab., Livermore, CA.

Pasquill, F., 1971: Atmospheric dispersion of pollution. *Quart. J. Roy. Meteoro. Soc*., **97**, 369.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 1993: *Numerical Recipes, 2nd. ed.*, Cambridge University Press, Cambridge, U.K.

Rodean, H.C., R. Lange, J.S. Nasstrom, and V.P. Gavrilov (1992) Comparison of two stochastic models of scalar diffusion in turbulent flow. Preprint, Tenth Symposium on Turbulence and Diffusion, 29 Sept.–2 Oct. 1992, Portland, OR. Published by the American Meteorological Society, Boston, MA.

Rodriguez, D.J., H. Walker, N. Klepikova, A. Kostrikov, and Y. Zhuk (1995) Evaluation of pollutant dispersion models over continental scales, *Atmos. Environ.*, **29**, 799-812. Also UCRL-JC-115003, LLNL, Livermore, CA.

Sasaki, Y., 1958: An Objective Analysis Based on the Variational Method. *J. Meteoro. Soc. Jap*., **36**, 77.

Sasaki, Y., 1970a: Some basic formalisms in numerical variational analysis. *Mon, Wea. Rev*., **98**, 875.

Sasaki, Y., 1970b: Numerical variational analysis formulated under the constraints determined by longwave equations and low-pass filter. *Mon. Wea. Rev*., **98,** 884.

Sherman, C.A., 1978a: MATHEW: A mass-consistent wind field model. Ph.D. Thesis, University of California, Davis/Livermore. Lawrence Livermore National Laboratory Report UCRL-52479.

Sherman, C.A., 1978b: A mass-consistent model for wind fields over complex terrain. *J. Appl. Meteoro.*, **17**, 312–319.

Slade, D.H. (ed.) (1968) *Meteorology and Atomic Energy*, U.S. Atomic Energy Commission (Available from NTIS, Springfield, VA as TID-24190)

Start, G.T. and L.L. Wendell (1974) *Regional Effluent Dispersion Calculations Considering Spatial and Temporal Meteorological Variations*, NOAA Tech. Memo. ERL ARL-44, Air Resources Lab., Idaho Falls, Idaho (Available from NTIS, Springfield, VA)

Sugiyama, G., R.E. Lee, and H. Walker, 1994: Conjugate Gradient MATHEW. Lawrence Livermore National Laboratory, UCRL-ID-118629.

Varga, R., 1962: *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, New Jersey.

Weil, J.C. and D.P. Hoult, 1973: A correlation of ground level concentrations of sulfur dioxide downwind of the keystone stacks. *Atmospheric Environment*, **7**, pp. 707–721.

Yanskey, G.R., E.H. Markee, Jr., and A.P. Richter (1966) *Climatography of the National Reactor Testing Station*, IDO-12048, Air Resources Field Research Office, National Reactor Testing Station, Idaho Falls, Idaho. (Available from NTIS, Springfield, VA)

Appendix B. TOPOG Interaction

The interactive capabilities in TOPOG provide a means to customize the model topography to the needs of a problem when the default elevation data averaging in TOPOG produces less than ideal results. These interactive capabilities are a combination of graphical presentations of the elevation data (color coded cells, contours, and cell values for a subset of the grid) and SmartStar menus/forms that permit the user to select the desired options. There are three menus in TOPOG: the MAIN MENU, the UNITS MENU, and the EDIT MENU. The main menu is the top level in the heirarchy with the other two menus accessed from the MAIN MENU. The UNITS MENU is selected via the 'Change modes' option on the MAIN MENU while the EDIT MENU is also selected via the 'Edit subgrid' option with the additional step of selecting a region on the display for more detailed examination with a graphics input device, typically a mouse. These menus are shown below.

TOPOG MAIN MENU
1. Edit subgrid
2. Fill holes
3. Change modes
4. Plot contours
5. Plot cell levels

TOPOG UNITS MENU
1. Units to DELZ
2. Units to METERS
3. Units to FEET
4. Set contour interval
5. Set DELZ
6. Reset grid base

TOPOG EDIT MENU
1. Edit single cell
2. Edit subgrid row
3. Edit subgrid column
4. Edit MATHEW row
5. Edit MATHEW column
6. Undo last edit

The follow is a discussion of the use of TOPOG interactive capabilities taken almost entirely from a series of documents written by John Pace. As these provide an excellent and complete description of the thoughtful use of these tools, his work is included here adapted slightly for the MATHEW/ADPIC User's Guide. First, each option in the TOPOG menus is discussed. After this a good example of the use of these features is provided.

When you first run TOPOG, you get a cell-level map of the entire MATHEW grid area on the display screen, and the MAIN MENU on the control screen.

The MAIN MENU contains the following options:

1. Edit subgrid. Selecting the "Edit subgrid" option from the TOPOG MAIN MENU means you want to edit a 10X10 subgrid of the full MATHEW grid. So, you have to tell the system where the subgrid is that you want to edit. On the full display map, you left-click where you want the center of your subgrid to be, and then (after the display map is redrawn, for reasons related to the GKS graphics standard) you left-click again to verify you really wanted the subgrid where you said you did. If you change your mind, you can select a different subgrid center by center-clicking. (If you put the center of the subgrid close to the edge of the MATHEW grid, the full size subgrid is built, but TOPOG invents subgrid cells for the area not covered by the MATHEW grid, with fictitious values in the subgrid cells that aren't really there.) The system draws your 10X10 subgrid. Then, when you switch to the control screen and hit return, you get the EDIT MENU, which is described below.

2. Fill holes. This is the fastest and easiest way to finish TOPOG and go on to the rest of the problem. You must be in DELZ mode to use this option (see the Change modes option).

3. Change modes. Enter the UNITS MENU.

4. Plot contours and use this display of the elevation data as the default. You can control the contouring interval (CONTOUR_INTERVAL) and/or the model's vertical grid spacing (DELZ) in the TOPOG.NML file, or interactively by using the Change modes option. This option lets you see a contoured display of the terrain field of the entire MATHEW grid. This function is controlled by the CONTOUR_INTERVAL and DELZ parameters, which can be set in the TOPOG.NML file or modified as discussed above, under the "Change modes" option. The contouring graphics package calculates how many contours it needs to use, with the interval set by CONTOUR_INTERVAL, to build the entire depth of the model grid (usually, 14 times DELZ for an R41 grid). The package uses only 7 colors, so if it needs more than 7 contour intervals it assigns several consecutive contour intervals to the same color. Thus it's possible to see several green contours, for example, surrounding a yellow contour. Once you've created a contour map, it's quick and easy to redraw the map using new contour control parameters (CONTOUR_INTERVAL and DELZ). From the MAIN MENU, select "Change modes" to get to the UNITS MENU. Then you can change "Set contour interval" and/or "Set DELZ". Hit GOLD/ZERO and you'll get a new contour map. It's a rather unusual graphics effect as the old map melts away--give it a try.

5. Plot cell levels and use this display of the elevation data as the default. This will return the default display map to what you started with when you first ran TOPOG.

The EDIT MENU contains the following options:

1. Edit single cell. This option lets you change the elevation value of just one subgrid cell. (This was the main subject of yesterday's note.) If you're in the DELZ mode, you assign a new elevation in DELZ units. If you're in the METERS or FEET modes, you assign a new actual elevation. When you select this option, you must tell TOPOG which cell in the subgrid is to be edited. So, left-click on the elevation value in the cell to be changed. The subgrid is redrawn with the elevation of that cell in yellow. Left-click again, somewhere on the subgrid, to verify your selection. Click on the control screen. Enter a new cell elevation value, in the units set by the mode you're in, and hit return. The elevation value of the cell is changed and is now displayed in white. Continue editing this subgrid by hitting keys 1-5, or hit GOLD/ZERO to get back to the MAIN MENU with the change retained, or hit 6 to cancel the edit and get back to the EDIT MENU.

2. Edit subgrid row. This option lets you edit all 10 cells in a subgrid row. You must tell TOPOG which row to edit. Left-click on an elevation value in any cell in the row you want to change. The subgrid is redrawn, with the values in all the cells in that row changed to yellow. Left-click again to verify your selection. Switch to the control screen. Enter the new values, in the units you're working in (DELZ units, meters, or feet), in the 10 available spaces. These new values go into the subgrid cells in your row, from left to right, when you hit GOLD/ZERO. You'll see the new values in white. Continue editing this particular subgrid, or hit GOLD/ZERO again to get back to the MAIN MENU and retain the change, or hit 6 to cancel the edit and get back to the EDIT MENU.

3. Edit subgrid column. Pretty much the same as "Edit subgrid row", except of course you're selecting a particular column of values to edit. When you enter new values on the control screen, keep in mind they'll be inserted from bottom to top of the subgrid.

4. Edit MATHEW row. This option lets you enter values for all the cells in a row of the MATHEW grid (usually 50 of them). You can see only 10 of the grid cells in your subgrid, so you shouldn't try to use this option for the purpose of eliminating holes. However, if you need to enter values for a grid, when you don't have terrain data to work with but must instead get elevations from a topographical map, this is the way to go.

5. Edit MATHEW column. Same as Edit MATHEW row for a column.

6. Undo last edit. As mentioned above, this option erases the effect of the last edit you made, and returns you to the EDIT MENU.

The UNITS MENU contains the following options:

1. Units to DELZ. When you set up the TOPOG.NML file, you either define DELZ explicitly as the vertical grid spacing, or you define the complete vertical depth of the grid (ZRANGE) and let TOPOG calculate DELZ for you by dividing ZRANGE by the number of vertical grid spaces (usually 14). Once DELZ is known, TOPOG sets up categories of DELZ units. For example, if DELZ=50 meters, the units of DELZ correspond to various height categories as follows:

DELZ =	1	2	3	4	etc.
ELEV Range (m) =	0-24	25-74	75-124	125-174	

A useful thing to remember is that if you've changed the mode to something other than DELZ, you can't select "Fill holes" from the MAIN MENU until you've changed the mode back to "Units to DELZ".

One problem with the DELZ units system is that it doesn't reveal much about the detailed vertical structure. As I discussed yesterday, you can see a lot more by switching the mode to METERS (or FEET). TOPOG defines one-cell holes as cells where the DELZ value is less than the DELZ value in the four surrounding cells. So, using the DELZ mode can help you see where holes may still exist, even after you've done some editing. Also, since TOPOG defines its output terrain in DELZ units, you can use the DELZ mode to see what TOPOG considers to be its output. After you've selected "Units to DELZ", hit GOLD/ZERO to return to the MAIN MENU.

2. Units to METERS. If you select this mode (and hit GOLD/ZERO to return to the MAIN MENU), you'll be able to see the actual elevation in meters that TOPOG has assigned to each cell when you edit a subgrid. And, when you enter new subgrid cell elevation values, they'll be in meters, not in DELZ units. It's not difficult to set up a table relating DELZ and the actual elevation in meters, as I did above, so it's easy to be sure you're changing heights enough to result in a new DELZ value, and thereby eliminate holes. Keep in mind that TOPOG will convert all heights into DELZ units before creating its output files, so you can't create more resolution than DELZ provides. But

the advantage of the "Units to METERS" option is that it lets you see what you need to know, so you can fill or expand holes effectively.

3. Units to FEET. Compared to "Units to METERS", this mode provides a little more detailed information about the actual elevation in each subgrid cell. However, it's harder to use, because it's harder to relate DELZ units (which are based on meters) to feet. So, it's less clear whether or not you've changed a cell's height enough to get it into the next DELZ category. Its main function is to facilitate in the input of elevation data directly from a contour map where the contour elevations are in feet.

4. Set contour interval. This value has no effect on the actual execution of the model. Its effect is only on the optional display you can get by selecting "Plot contours" from the MAIN MENU (see below). This option effectively allows you to override the choice of CONTOUR_INTERVAL you made in TOPOG.NML. Or, if you didn't define CONTOUR_INTERVAL in TOPOG.NML, TOPOG will use the DELZ value as the contour interval unless you use the "Set contour interval" option. Regardless of whether you're in DELZ, METERS, or FEET mode, when you set the contour interval you're entering a value in meters!

5. Set DELZ. In contrast to the "Set contour interval" option, this option has a strong effect on the actual execution of TOPOG. It lets you change the actual vertical structure of the model grid! If you change DELZ, TOPOG will recalculate all the results that depend on vertical structure, including which cells are now holes. As with "Set contour interval", the value you enter for DELZ is always in meters, regardless of what mode you're working in.

6. Reset grid base. Substantial editing of the cell heights can cause the base of the model topography to be offset. The grid base is adjusted during the averaging process so the bottom of the grid volume is at the zero cell height index (corresponding to an index of one in MEDIC and MATHEW). If editing causes the lowest level of the model topography to be raised above zero, then the volume of the grid above the terrain is reduced unnecessarily. The 'reset grid base' option permits the grid base to be shifted so the lowest index is again zero. An important example of the use of this option occurs when a complete elevation grid is input interactively based on values read from a contour map. Typically, TOPOG is initiated with FLAT_TOPO set to 'Y' setting the initialize cell heights to zero. The grid base is also set to zero. After the cell heights have been entered, the grid base remains at zero, while the cell heights may all extend beyond the grid top (e.g., consider entering elevations from the Himalayas with a 100 m DELZ producing a minimum cell height of, perhaps, 40 or 50). In such a case, 'reset grid base' must be performed to produce a grid that includes some atmosphere.

Interactive example:

As mentioned in Section II.C.1, single cell holes in the model topography must be avoided. These can be removed automatically, either by running TOPOG with INTERACTIVE set to 'N' or by selecting option 2 in the MAIN MENU A second way to eliminate holes in the initial TOPOG terrain field, that in some cases may be better than the default hole filling, is to edit the cell heights interactively. This approach lets you eliminate holes while changing the terrain field as little as possible, and/or retaining as much as possible the significant features of the terrain. The drawback is that it takes longer.

The terrain files have 1-meter-resolution elevation data, and TOPOG's job is to convert these values into representative heights, in multiples of DELZ, for each MATHEW/ADPIC grid cell. For example, if you've set DELZ=50 m, the units of DELZ correspond to heights in various categories, as shown:

DELZ =	1	2	3	4	etc.
ELEV Range (m) =	0 - 24	25 - 74	75 - 124	125 - 174	
Avg Elev (m) =	12	50	100	150	

So, if the elevation in a grid cell is between (and including) 25 and 74 meters, TOPOG will assign it a DELZ value of 2.

A one-cell hole has a DELZ value (corresponding to these ranges of actual elevations) at least one DELZ unit less than the values in the four rook's case neighboring cells. If you let TOPOG fill the hole automatically, it will increase the hole's DELZ value so it matches the lowest elevation in any of the surrounding cells.

You have the capability to fill holes manually. If you choose to do this, you'll see the system is set up to show each cell's elevation in DELZ units, with one-cell holes identified as red squares. You have two ways to eliminate one-cell holes: by increasing the cell-hole's DELZ value to match or exceed the surrounding values; or by reducing the height of an adjacent grid cell to create a larger valley than you had before. Either approach satisfies ADPIC. But this configuration gives you no way to know which of these options is better (more faithful to the actual terrain) in a given situation. However, TOPOG can provide additional information to assist in resolving the ambiguity The assessor can look at the actual elevation in meters in each cell, and make an educated decision as to whether the hole should be filled or the valley should be expanded.

To do this alternate procedure, do the following:

a. Start TOPOG in the ordinary way. You'll see the usual map, with black squares showing the locations of all the holes.

b. On the MAIN MENU, select item 3 (Change modes). On the UNITS MENU, select item 2 (Units to METERS). Then press GOLD ZERO to get back to the MAIN MENU. Now press 1 (Edit subgrid). Move the cursor near the hole you want to eliminate, and left-click. After the screen settles down, left-click again to accept that location (or center-click if you want to pick another location). The screen is redrawn as a 10 X 10 subgrid, with each cell's elevation shown in meters. Now you have the information you need to make the best choice of whether a hole will be filled or expanded.

c. You make the change by altering one or more cell's elevation so it crosses into the next DELZ category (either up or down). To do this, go back to the control screen. The EDIT MENU should be active. Select item 1 (Edit single cell). Left-click your cursor on the cell you want to change. When the screen settles down, left-click again to verify your choice. You'll see the elevation value is now displayed in yellow. Now click on the control screen, and enter the new elevation value for the cell. The new elevation is shown in white on the subgrid cell. Hit GOLD ZERO to exit, retaining the change. If you've eliminated one or more holes, you'll see that reflected in the screen display.

Here's an example. I made a TOPOG run which initially had 11 one-cell holes in the grid. The grid cells in the vicinity of one of the holes had these elevations (in meters):

54	58	32	55	52	33
63	57	18	49	58	28
67	46	25	49	44	32
62	28	43	38	19	42

The one-cell hole was the "18" value on the second line. Notice that there was a "25" value immediately south of the hole, so there actually was an area of lower elevation in these two cells, surrounded by higher terrain. If the "25" was instead a "24", TOPOG wouldn't think this was a one-cell hole at all. (The "19" value on the fourth line was north of another lower value, so it wasn't a one-cell hole.)

The DELZ values corresponding to this initial field were:

2	2	2	2	2	2

2 2 1 2 2 2

2 2 2 2 2 2

2 2 2 2 1 2

If I had let TOPOG fill the holes automatically, the "1" on the second line would have become a "2", and the resulting elevation field wouldn't have reflected the lower terrain in that area. On the other hand, it was easy to change the "25" value to "24" (while in the meters mode), eliminating the one-cell hole while retaining the lower-elevation feature. (Or, I could have changed that cell's "2" to a "1" while I was in the DELZ mode). The point is that if you don't use the meters mode, you won' t know which approach will best preserve the terrain's characteristics. And, of course, if you let TOPOG fill holes automatically you can't make these decisions at all. This hands-on procedure takes longer than letting TOPOG do it automatically. So, perhaps you might want to use it only for holes close to the expected path of the plume. Then you can let TOPOG fill the rest of the holes on its own.

Appendix C. Traceback Block Description

Traceback block structure

Historically, the transfer of parameters and the traceback functions was implemented via an ID block, which was intended to be a sequential list of all model parameters and meteorological data used in a model execution and was placed at the beginning of each file used to transfer data between models. Each such file contains space for a complete ID block which with appropriate sections being filled in as the model sequence was executed. As a result, much of thespace was not utilized until the completion of ADPIC.

While the ID block functioned for 15+ years, it's use implied a number of constaints that limited the development of the models and increased the difficulty of maintenance. Many of these contraints were associated with the "hard-wired" aspects of this mechanism that forced recompilation of the entire model stream to make any change in the ID block structure along with an inability to use any binary files generated with an old ID block. As a result, there was a strong tendency to avoid changes in the modeling system. Such inertia is inappropriate for an active model development environment. Therefore, it was necessary to replace the ID block with a new model environment that supports the necessary parameter passing and traceback while permitting easier changes to the models.

The traceback block (TBB) is composed of series of sections, one for each model plus an additional section for general pointers that define the modeling environment and a one entry section giving the name of the current file. The location of individual entries within a TBB section is specified by parameters defined in an INCLUDE file as follows:

TBB_SELF.CON

TBB_GEN.CON

TBB_TOP.CON

TBB_MED.CON

TBB_MAT.CON

TBB_ADP.CON

TBB_PLC.CON

TBB_TIM.CON

New files requiring traceback support are to be added at the end of the appropriate section (the mechanism for defining section lengths is discussed under the next heading in this document). Thus, the order of traceback entries, once defined, must remain unchanged to support the use of older data files with newer models. Adding a new file entails updating the INCLUDE file and recompiling the models that make use of the TBB section. An unused entry will be maintained at the end of each section to support testing of a model with an additional file without recompiling other models that don't read the new file.

The following list is the current structure of the TBB in order. Each entry is a 40 character string. Note that directory names are assigned two entries. Also note that .PRM extensions indicate files in the Model Parameters System:

Self 1 current file name
General
1 MODEL$A_GRID
3 MODEL$A_WINDS
5 MODEL$A_DISPERSION
7 MAPS$A_REGIONAL
9 MAP$A_PROJECTION_PARAMETERS
10 SITE_ID
11 PROBLEM_ID
12 TOPOG_GRID
13 MAP$F_SITEMAP
14 MAP$F_PROJECTION_PARAMETERS
15 Unused Entry
16 Unused Entry

TOPOG
1 TOPOG.NML.CNML
2 TOPOG.PRM
3 TOPO$A_REGIONAL
5 TOPO$F_REGIONAL
6 unused entry

MEDIC
1 MEDIC.CNML

2 STNLOC.CMET
3 OBSERV.CMET
4 MEDIC.PRM
5 REGMET.PRM
6 GWC Grid directory
8 GWC Grid
9 RWM Grid directory
11 RWM Grid
12 unused entry
13 unused entry

MATHEW
1 MATHEW.CNML
2 MATHEW.PRM
3 REGMET.PRM
4 MEDVEL.BIF
5 unused entry

ADPIC
1 ADPIC.CNML
2 ADPIC.RST
3 ADPIC.PRM
4 REGMET.PRM
5 First MEDVEL.BIF/MATVEL.BIF
6 Last MEDVEL.BIF/MATVEL.BIF
7 ADPIC.VLST
8 HYBRID_DB
9 ADPIC.RST
10 CONCRST
11 RESUS.BIF (unused)
12 unused entry
13 unused entry

PLCNT
1 PLCNT.CNML
2 CONTOUR.PRM

3 DOSECON.PRM
4 First CONC.BIF
5 Last CONC.BIF
6 PLCNT.CLST
7 unused entry

TIMEHIS
1 TIMEHIS.CNML
2 TIMEHIS.PRM
3 DOSECON.PRM
3 SAMPLER.CDAT
4 First CONC.BIF
5 Last CONC.BIF
6 TIMEHIS.CLST
7 unused entry

Note that the first three entries in the general section are three logical names that control the directories where data files are found. MODEL$A_GRID specifies the directory where the TOPOG*.GRID file is found. MODEL$A_WINDS indicates the location of the data used by MEDIC and MATHEW. And MODEL$A_DISPERSION shows where ADPIC, PLCNT and TIMEHIS data can be found. This permits reuse of previously created TOPOG*.GRID and MED/MATVEL files without recopying the files to a new run directory. Also note that this same structure is used for the hemispheric models that are essentially a parallel model stream in terms of input files with the exception of HMEDIC.

Traceback block structure definition

It can be assumed with some confidence that the ARAC models will continue to require new data files over time in order to support new data sources and more sophisticated modeling functions. These new files will require traceback support. To facilitate such support and to avoid incompatibilities in the use of old model files, a capability has been developed to allow old model files to describe their traceback structure via metadata at the beginning of the TBB. This involves maintaining a file listing the current number of entries in each TBB section

(TBB_STRUCTURE.CON). This information is placed at the beginning of each .GRID and .BIF file thus defining the TBB structure that was in use when the file was created. The structure definition includes the number of sections in the TBB, thereby permitting addition of new models and other programs to the traceback system. When an old file is read, the metadata will allow later codes to determine the location of the necessary filenames inspite of the out-of-date TBB structure.

All the models that read the TBB have buffers of TBB entries that are long enough to hold the current TBB plus room for a substantial amount of growth to minimize the need to recompile the codes. As the ARAC models develop, these buffers will gradually be used more completely. If they are ever nearly full, they would have to be expanded; thus, they can never get shorter only longer. Increasing the buffer size may be an inconvience but it should be simple to perform and will not cause an inability to read and process old data files.

Because the various buffers only grow with time and each .GRID and .BIF file defines its own TBB structure, any files created using this system will be readable by any future version of the MATHEW/ADPIC models for which the basic model and grid structure remains consistent (a terrain-following coordinate MATHEW/ADPIC or a finite-element SABLE/ADPIC would presumbly imply an unavoidable discontinuity in the file handling system). Thus, it should be possible to use an old .GRID file with a antiquated TBB and wind grids using a more recent TBB with the latest version of ADPIC without difficulty. The ease of updating the TBB structure suggests that it is not necessary to leave large amounts of empty space within the current TBB to support future model development, although the TBB buffers within the codes should support an appropriate amount of extra space.

A sample TBB header follows:

Self ID
General
TOPOG
MEDIC
MATHEW
ADPIC
PLCNT
TIMEHIS

Appendix D. ARAC Model System Logical Names

VAX VMS logical names play a central role in the ARAC modeling system and it is important for model users to have some familiarity with these tools (it is intended that most or all of the functionality implemented in VMS with logical names will mapped to environment variables in Unix versions of the system). Logical names are used to point at all directory paths in file handling operations and to some files. In doing this, flexibility is provided in locating input and output files in a directory structure that would not be possible if the disk/directory paths were set explicitly in the codes. There are several logicals that are used throughout the model system. In addition, TOPOG, MEDIC and PLCNT have a number of additional logicals that define connections to various parts of the system.

General Logicals

There are four logicals that are central to the execution of the ARAC models. The first three control the directories where input and output from the various codes is accessed. MODEL$A_GRID is associated with TOPOG, MODEL$A_WINDS is associated with MEDIC and MATHEW, and MODEL$A_DISPERSION is associated with ADPIC, PLCNT and TIMEHIS. More specifically, TOPOG reads its control input (TOPOG.NML) and writes the TOPOG*.GRID file to the MODEL$A_GRID directory. All subsequent models also read the TOPOG*.GRID file from this directory. Because this directory specification is a logical name the various models can read the TOPOG*.GRID file from another directory produced as part of a different run thereby avoided the need to copy the TOPOG*.GRID file from run directory to run directory to execute different model runs using the same grid geometry.

MEDIC and MATHEW read their control input (MEDIC.NML, OBSERV.MET, STNLOC.MET, MATHEW.NML and MEDIC.VLST) and write their output to MODEL$A_WINDS. ADPIC, PLCNT and TIMEHIS read their control input (ADPIC.NML, CONC_ANALYSIS.NML, ADPIC*.CLST and SAMPLER.DATA) and write their output to MODEL$A_DISPERSION. These logicals are set by the restructured model version of SELRUN in the operational system and in the exem environment. *These logicals must be set properly for the system to run correctly*.

The fourth essential logical is MODEL$F_GRID that defines the location of the TOPOG*.GRID file. This logical typically incudes MODEL$A_GRID: followed by the fully qualifer TOPOG*.GRID filename. MODEL$F_GRID is defined by TOPOG when the code is executed. Thus, in a straightforward execution of the models, TOPOG is run, followed immediately by MEDIC with no operation being required to set this logical name. When returning to a directory or when a TOPOG*.GRID file is to be used from a different run directory, then the SELGRI command is run that allows the user to select a MODEL$A_GRID directory along with the appropriate TOPOG*.GRID file. *This logical must also be set for the system to run correctly*.

Other logicals in general use are GEOG$A_REGIONAL and GEOG$F_REGIONAL that point to the geography file that is to be used as a base map for various ARAC model plots. GEOG$A_REGIONAL points at the directory that contains a map family while GEOG$F_REGIONAL consists of GEOG$F_REGIONAL: followed by the specific map file name. These logicals are not essential to the running of the models, if a map file cannot be opened and read successfully, the models just don't plot a basemap. The ARAC system logicals SITE_ID, PROBLEM_ID and RUN_ID are translated and placed in the traceback block to provide a record of the run environment when a file was generated.

TOPOG Logicals

TOPOG relies on two logicals that define the elevation data to be used. These are TOPO$A_REGIONAL and TOPO$F_REGIONAL with GEOG$A_REGIONAL point to the directory that contains the elevation data files and TOPO$F_REGIONAL consisting of TOPO$F_REGIONAL: followed by the specific elevation file name. These logicals must be set if FLAT_TOPO is 'Y' in TOPOG_CONTROL_NML but are not used if FLAT_TOPO is "N'.

Map Projection Logials

The models are executed on the current map projection as defined by the logical MAP$F_PROJECTION_PARAMETERS. This logical points at an ASCII file that defines a projection. The pathname for the projection file is defined by the logical MAP$A_PROJECTION_PARAMETERS. Translations of both these logicals are entered into the traceback block by TOPOG when a .GRID file is created. Note that the value in MAP$A_PROJECTION_PARAMETERS and MAP$F_PROJECTION_PARAMETERS apply problem-wide, that is, all runs in a problem use the same map projection. Outside of the operational system, the user must ensure that the MAP$F_PROJECTION_PARAMETERS and MAP$A_PROJECTION_PARAMETERS logicals are set to point at the appropriate projection definition file.

MEDIC Logicals

MEDIC utilizes a number of logicals in order access meteorological information from various sources. The problem station library is the automated source of meteorological observing station locations (thus, superceding the STNLOC.MET file) and is accessed using the logical EDITPSL$F_PROBLEM_STATION_LIBRARY. This information may also reside in a global section that is accessed with GBL$EDITPSL$G_PSL. Metdata can be read from a problem metdata file (thus replacing the OBSERV.MET file) in the directory specified by MAKEPMF$A_PROBLEM_METDATA_ARCHIVE. These logicals do not have to set if observing locations and metdata are read from a STNLOC.MET and an OBSERV.MET file (see MEDIC.NML description for the appropriate control variables). The central longitude and hemispheric indicator for transforming UTM map projection information (i.e., central longitude and hemisphere indicator) is currently found in the questionnaire that is accessed with the two logicals: ENTQUEST$F_QUESTIONNAIRE_CHANGE_JOURNAL and

ENTQUEST$F_QUESTIONNAIRE_ITEM. These two logicals do not need to be set if MERIDIAN and HEMI_NS are set in MEDIC_CONTROL_NML.

PLCNT Logicals

A number of logicals are used by PLCNT to complete the descriptive information for the products and to handle the distribution of products in the ARAC system. This descriptive information relies on the logicals ARAC$USER_NAME to determine the account used to generate the product and the SITE_ID, PROBLEM_ID and NODE_ID to determine the execution environment. The site workstation model products are placed in the directory specified by MODEL_PROD$A_PRODUCTS. The product description records for the site workstation products are written to MODEL_PROD$F_SWS_DESCRIPTIONS. The frame count (to ensure, when combined with the generation date in the product filename, that each product name is unique over the entire operational ARAC system) is maintained for the site workstation products in MODEL_PROD$F_SWS_FRAME_COUNT.

The complete association of directories and files with logical names is listed as follows:

TOPOG.NML	MODEL$A_GRID	
TOPOG*.CNML	MODEL$A_GRID	
TOPOG*.CGMB	MODEL$A_GRID	
TOPOG* .GRID	MODEL$A_GRID	MODEL$F_GRID
TOPOG.LOG	MODEL$A_GRID	
regional elevations	TOPO$A_REGIONAL	TOPO$F_REGIONAL
MEDIC.NML	MODEL$A_WINDS	
MEDIC*.CNML	MODEL$A_WINDS	
STNLOC.MET	MODEL$A_WINDS	
STNLOC*.CMET	MODEL$A_WINDS	
problem stations		EDITPSL$F_PROBLEM_STATION_LIBRARY
OBSERV.MET	MODEL$A_WINDS	
OBSERV* .CMET	MODEL$A_WINDS	
problem metdata	MAKEPMF$A_PROBLEM_METDATA_ARCHIVE	
GWC grids	DECAFGWC$A_AFGWC_GRID_FILE	
MEDIC*.CGMB	MODEL$A_WINDS	
MEDVEL*.BIF	MODEL$A_WINDS	
MEDIC.LOG	MODEL$A_WINDS	
MEDIC.VLST	MODEL$A_WINDS	
questionnaire		ENTQUEST$F_QUESTIONNAIRE_ITEM
MATHEW.NML	MODEL$A_WINDS	

MATHEW* .CNML	MODEL$A_WINDS
MATHEW* .CGMB	MODEL$A_WINDS
MATVEL*.BIF	MODEL$A_WINDS
MATHEW.LOG	MODEL$A_WINDS
MATHEW.VLST	MODEL$A_WINDS

ADPIC.NML	MODEL$A_DISPERSION
ADPIC* .CNML	MODEL$A_DISPERSION
ADPIC*.RST	MODEL$A_DISPERSION
ADPIC* .CGMB	MODEL$A_DISPERSION
CONC* .BIF	MODEL$A_DISPERSION
CONC* .VLST	MODEL$A_DISPERSION
CONC* .RST	MODEL$A_DISPERSION
PARTPOS* .BIF	MODEL$A_DISPERSION
ADPIC.LOG	MODEL$A_DISPERSION
ADPIC.CLST	MODEL$A_DISPERSION
MEDIC.VLST	MODEL$A_VLST
MATHEW.VLST	MODEL$A_VLST
CONC_ANALYSIS.NML	MODEL$A_DISPERSION
PLCNT* .CNML	MODEL$A_DISPERSION
PLCNT* .CGMB	MODEL$A_DISPERSION
PLCNT*.CLST	MODEL$A_DISPERSION
PLCNT_* .TBB	MODEL$A_DISPERSION
PLCNT.LOG	MODEL$A_DISPERSION
user name	ARAC$USER_NAME
SWS products	MODEL_PROD$A_PRODUCTS
SWS product descriptions	MODEL_PROD$F_SWS_DESCRIPTIONS
SWS frame count	MODEL_PROD$F_SWS_FRAME_COUNT.

(SWS = Site Workstation System)

Appendix E. ARAC Model Namelist Summary

The following tables contain alphabetically sorted lists of all namelist parameters for each model, including their dimensions and input types (real, integer,character). The MEDIC table also indicates the namelist variables that are used in hemispheric as well as regional MEDIC. Array dimensions are given in terms of parameters which are listed in the first table which along with their default values.

DIMENSION PARAMETERS	*DEFAULT VALUE*	*FILE*	*MODEL*
EXPL_CLD_DIM	9	adpic_params.con	adpic
IMAXMAT	51	dims_41_41_15.con	modgrid
JMAXMAT	51	dims_41_41_15.con	modgrid
KMAXMAT	15	dims_41_41_15.con	modgrid
KMAX_UPR_NODDS	9	nodds_dims.con	modutil
KMAX_UPR_HEMI	15	hemi_dims.con	modutil
KMAX_UPR_RWM	10	rwm_dims.con	modutil
MAXCONTLEVELS	5	conc_analysis_dims.con	plcnt
MAXDBINS	30	conc_analysis_dims.con	plcnt
MAX_EXPL_CLD_PROF	30	adpic_params.con	plcnt
MAXFACTORS	15	timehis_dims.con	timehis
MAXFBINS	30	conc_analysis_dims.con	plcnt
MAXHANGLES	10	timehis_dims.con	timehis
MAXLEVEL_ALL	35	upperair_dims.con	medic
MAXPLOTSERIES	5	conc_analysis_dims.con	plcnt
MAXPROF	10	profile_dims.con	modutil
MAX_RCPTR	500	adpic_receptor_dims.con	adpic
MAX_RESTART	10	adpic_dims.con	adpic
MAXSAMPLERS	400	timehis_dims.con	timehis
MAXSBINS	30	adpic_dims.con	adpic
MAXSOURC	9	adpic_dims.con	adpic
MAXTBINS	15	timehis_dim.con	timehis
MAXTIM	100	adpic_params.con	adpic
MAX_TIMES	10000	MATHEW_specific_params.con	mathew
MAXTIMESTEPS	200	timehis_dims.con	timehis
MAX_TRAJ	500	adpic_receptor_dims.con	adpic
MAX_TRAJ_TIMES	100	adpic_receptor_dims.con	adpic
MAX_VEL	10000	vel_dims.con	modutil
NANGLESTEPS	11	timehis_dims.con	timehis

NAMELIST VARIABLES	*TYPE*	*DIMENSIONS*
TOPOG_CONTROL:		
CONTOUR_INTERVAL	real	1
FLAT_TOPO	char*1	1
INTERACTIVE	char*1	1
LISTDETAILS	char*1	1
RESTART	char*1	1
TOPOG-PARAMS:		
DELX	real	1
DELY	real	1
DELZ	real	1
DOUBLECELL	char*1	1
MAP_ADJUSTMENT	char*1	1
RANGE	real	1
X_GRIDCENTER	real	1
X_OFFSET_MAT	real	1
X_ORIGIN_PIC	real	1
XRANGE	real	1
Y_GRID_CENTER	real	1
Y_OFFSET_MAT	real	1
Y_ORIGIN_PIC	real	1
YRANGE	real	1
ZRANGE	real	1

NAMELIST VARIABLES	*TYPE*	*DIMENSIONS*	*SPECIAL*
MEDIC_CONTROL:			
GWC_BARB_PLOT_FREQ	integer	1	MEDIC/HMEDIC
GWC_BARB_SCALE	real	1	MEDIC/HMEDIC
HORZ_VECTOR_SCALE	real	1	MEDIC/HMEDIC
IPROFILE	integer	MAXPROF	MEDIC/HMEDIC
JPROFILE	integer	MAXPROF	MEDIC/HMEDIC
MEDIC_VL:ST	char*6	1	MEDIC/HMEDIC
PLOT_LEVELS	integer	KMAXMAT	MEDIC/HMEDIC
PLOT_LEVELS_NHE	integer	KMAX_UPR_HEMI	MEDIC/HMEDIC
PLOT_LEVELS_RWM	integer	KMAX_UPR_RWM	
PLOT_LEVELS_SHE	integer	KMAX_UPR_HEMI	MEDIC/HMEDIC
PRINT_WINDFIELDS	char*1	1	MEDIC/HMEDIC
REGMET_PARAMFILE	char*22	1	
RWM_BARB_PLOT_FREQ	integer	1	MEDIC/HMEDIC
RWM_BARB_SCALE	real	1	MEDIC/HMEDIC
SFC_BARB_PLOT_FREQ	integer	1	
SFC_BARB_SCALE	real	1	
SFC_ROUGH_HGT	real	1	
STNLOCPSL	char*1	1	

TERRAIN_ADJUSTMENT	char*1	1	MEDIC/HMEDIC
VECTOR_PLOT_FREQ	integer	1	MEDIC/HMEDIC
XPROFILE	real	MAXPROF	MEDIC/HMEDIC
YPROFILE	real	MAXPROF	MEDIC/HMEDIC
NAMELIST VARIABLES	*TYPE*	*DIMENSIONS*	*SPECIAL*

MEDIC_PARAMS:

ALL_WINDS_DIR	real	MAXLEVEL_ALL	MEDIC/HMEDIC
ALL_WINDS_HGT	real	MAXLEVEL_ALL	MEDIC/HMEDIC
ALL_WINDS_SPD	real	MAXLEVEL_ALL	MEDIC/HMEDIC
AVGTIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
BL_HGT	real	1	
EXTRAP_EXPNT_BL	real	1	
MAX_VEER_PARAM_VERT	integer	1	
MAX_VEER_PROF_HORZ	integer	1	
METDATASET	char*40	1	
MET_START_DATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
MET_START_TIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
NHEDATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
NHEFCST	integer	1	MEDIC/HMEDIC
NHETIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
PROBMET_DST	char*9	1	
PROBMET_DST_SFC	char*9	1	
PROBMET_DST_TOW	char*9	1	
PROBMET_DST_UPR	char*9	1	
PROFILE_EXTRAP	char*1	1	
PWR_LAW_EXPNT_SL	real	1	
REF_HGT	real	1	
RWMDATE	char*(LEN_DATE_STRING)	1	
RWMFCST	integer	1	
RWMTIME	char*(LEN_TIME_STRING)	1	
SHEDATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
SHEFCST	integer	1	MEDIC/HMEDIC
SHETIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
SINGLE_PROFILE_WINDS	char*8	1	
SL_HGT	real	1	
USE_UPR_AT_SFC	char*1	1	

NAMELIST VARIABLES	*TYPE*	*DIMENSIONS*
MATHEW_CONTROL		
HORZ_VECTOR_SCALE	real	1
IPROFILE	integer	MAXPROF
ISLICE	integer	IMAXMAT
ITER_MESSAGE	integer	
JPROFILE	integer	MAXPROF
JSLICE	integer	JMAXMAT
OUTPUT_DESTINATION_DIR	char*10	1
PLOT_LEVELS_ABOVE_TOPO	integer	KMAXMAT
PLOT_GRID_LEVELS	integer	KMAXMAT
PRINT_ADJUSTWIND_ADPIC	char*1	1
PRINT_ADJUSTWIND_DIVERG	char*1	1
PRINT_ADJUSTWIND_FACE	char*1	1
PRINT_ADJUSTWIND_GRID	char*1	1
PRINT_CONVERGENCE	char	1
PRINT_ICCG_CHECK	char*1	1
PRINT_INITIALWIND_DIVERG	char*1	1
PRINT_INITIALWIND_FACE	char*1	1
PRINT_INITIALWIND_GRID	char*1	1
PRINT_LAMBDAS	char*1	1
PRINT_SIDES	char*1	1
VECTOR_PLOT_FREQ	integer	1
VLST_OPTION	char*6	1
VERT_VECTOR_SCALE	real	1
XPROFILE	real	MAXPROF
XSLICE	real	IMAXMAT
YPROFILE	real	MAXPROF
YSLICE	real	JMAXMAT
MATHEW_PARAMS:		
CONV_TOLERANCE	real	1
ELLIPTIC_EQN_TEST	char*1	1
FIX_LATERAL_INFLOW	char*1	1
FIX_XMAX_INFLOW	char*1	1
FIX_XMIN_INFLOW	char*1	1
FIX_YMAX_INFLOW	char*1	1
FIX_YMIN_INFLOW	char*1	1
ITER_HALT	integer	1
LATERAL_BOUNDARY	char*15	1
MEDVEL_NAMES	char*(TBB_STRING_LEN)	MAX_VEL
MIXED_LATERAL_ADJUST	real	1
MIXED_TOP_ADJUST	real	1
MIXED_XMAX_ADJUST	real	1

MIXED_XMIN_ADJUST	real	1
MIXED_YMAX_ADJUST	real	1
MIXED_YMIN_ADJUST	real	1
PARAMS_DATE	char*(LEN_TIME_STRING)	MAX_TIMES
NAMELIST VARIABLES	***TYPE***	***DIMENSIONS***
PARAMS_TIME	char*(LEN_DATE_STRING)	MAX_TIMES
SIGMA_HORZ	real	1
SIGMA_VERT	real	MAX_TIMES
STAB_MATHEW	integer	MAX_TIMES
TOP_BOUNDARY	char*15	1
XMAX_BOUNDARY	char*15	1
XMIN_BOUNDARY	char*15	1
YMAX_BOUNDARY	char*15	1
YMIN_BOUNDARY	char*15	1

NAMELIST VARIABLES

ADPIC_CONTROL:	***TYPE***	***DIMENSIONS***
DECAY_MODE	char*16	1
DIFF_METHOD	char*5	1
INITIAL_TIME_STEP	real	1
NESTS	integer	1
PARTPOS	char*1	1
PLOT_INTERVAL	char*(LEN_DUR_STRING)	1
PRINT_ADVECT_VEL	char*1	1
PRINT_CELL_CONC	char*1	1
PRINT_FRCTN_VEL	char*1	1
PRINT_MC_DIAG	char*1	1
PRINT_PART_COORD	char*1	1
PRINT_PART_VEL	char*1	1
PRINT_SAMPLING_BINS	char*1	1
PRINT_TOPOG	char*1	1
PRINT_TURB_PARAM_GRADI	char*1	1
PROBLEM_TITLE	char*70	1
RESTART_DATE	char*(LEN_DATE_STRING)	1
RESTART_TIME	char*(LEN_TIME_STRING)	1
RESTART_WRITE_DATE	char*(LEN_DATE_STRING)	MAX_RESTART
RESTART_WRITE_INTERVAL	char*(LEN_DUR_STRING)	1
RESTART_WRITE_TIME	char*(LEN_TIME_STRING)	MAX_RESTART
REFLECT_AT_MLHGT	char*1	1
RUN_DURATION	char*(LEN_DUR_STRING)	1
RUN_START_DATE	char*(LEN_DATE_STRING)	1
RUN_START_TIME	char*(LEN_TIME_STRING)	1
RUN_STOP_DATE	char*(LEN_DATE_STRING)	1
RUN_STOP_TIME	char*(LEN_TIME_STRING)	1
SAMP_CELL_DELZ_FRACT	real	1
SAMPLING_HGT	real	MAXSBINS
SAMPLING_HGT_REF	char*11	1
SAMPLING_INTERVAL	char*(LEN_DUR_STRING)	1
SAMPLING_TYPE	char*6	MAXSBINS
SKIP_RANDOM_COUNT	integer	1
SOURCES_TO_SAMPLING_BIN	integer	MAXSOURC,MAXSBINS
STATUS_INTERVAL	char*(LEN_DUR_STRING)	1
USE_EXPL_CLD_RISE	char*1	1
USE_MOVING_RECEPTORS	char*1	1
USE_VLST	char*1	1
WIND_TYPE	char*9	1
X_NEST_LOC	real	1
Y_NEST_LOC	real	1
X_WINDOW_MIN	real	1

X_WINDOW_MAX	real	1
Y_WINDOW_MIN	real	1
Y_WINDOW_MAX	real	1
Z_WINDOW_MIN	real	1
Z_WINDOW_MAX	real	1

ADPIC_METPARAMS:	***TYPE***	***DIMENSIONS***
DIFF_ON_X	char*1	1
DIFF_ON_X	char*1	1
DIFF_ON_Z	char*1	1
GAUSS_DIFF_MULT_X	real	1
GAUSS_DIFF_MULT_Z	real	1
HORZ_TIME_SCALE	real	1
INTERP_VEL_WINDS	char*8	1
INTERP_WINDS_STEP_MEAN	char*1	1
INV_OBUKHOV_LEN_ML	real	MAXTIM
INV_OBUKHOV_LEN_SL	real	MAXTIM
K_VON_KARMAN	real	1
KZ_CONSTANTZ	real	1
KZ_LINEARZ_REF	real	1
KZ_LINEARZ_REF_HGT	real	1
KZ_SIMTHRY_PHI_A	real	1
KZ_SIMTHRY_PHI_B	real	1
KZ_SIMTHRY_C	real	1
KZ_SIMTHRY_TROPO	real	1
MAX_DELZ_SHIFT	integer	1
MAX_DITHER	real	1
MAX_GLOBAL_DELTA_T	real	1
MET_START_DATE	char*(LEN_DATE_STRING)	MAXTIM
MET_START_TIME	char*(LEN_TIME_STRING)	MAXTIM
MIN_MASS_FRACT	real	MAXTIM
ML_HGT	real	MAXTIM
PWRLAW_EXPNT_U	real	1
PWR_LAW_EXPNT_SL	real	MAXTIM
PRECIP_SCAVENG_COEF	real	MAXTIM
PRECSCAV	real	1
RAIN_START	char*(LEN_DUR_STRING)	1
RAIN_STOP	char*(LEN_DUR_STRING)	1
SFC_ROUGH_HGT	real	1
SIGH_PWRLAWX_ALPHA	real	1
SIGH_PWRLAWX_BETA	real	1
SIGH_PWRLAWX_XMAX	real	1
SIGMA_THETA_FACT	real	MAXTIM
SIGZ_PWRLAWX_GAMMA	real	1
SIGZ_PWRLAWX_RHO	real	1

SIGZ_PWRLAWX_XMAX	real	1
STAB_CLASS	integer	MAXTIM
TIMESTEP_GLOBAL_FRACT	real	1
TIMESTEP_LOCAL_MAX	real	1
TIMESTEP_LOCAL_MIN	real	1
TIMESTEP_LOCAL_FRACT	real	1
TURB_PARAM_TYPE	integer	MAXTIM
TURB_PARAM_VERT	char*22	MAXTIM
NAMELIST VARIABLES	***TYPE***	***DIMENSIONS***
TURB_PARAM_HORZ	char*22	MAXTIM
U_CALM	real	1
U_MEAN_DIR	real	1
U_MEAN_HGT	real	1
U_MEAN_REF	real	1
U_MEAN_REF_HGT	real	1

ADPIC_RADPARAMS	**TYPE**	**DIMENSIONS**
DCF	real	500
DCF_NUCLIDE_NAME	char*20	500
DCF_PATH_DESCRIPTOR	char*20	500
DECAY_START_INTERVAL	char*13	9
DOSE_PATH_DESCRIPTOR	char*20	1
NUCLIDE_MIX	real	100
NUCLIDE_NAME	char*20	100
NUMBER_OF_TOXIC_TIMES	integer	1

ADPIC_RECEPTOR	**TYPE**	**DIMENSIONS**
RCPTR_DELX	real	MAX_RCPTR
RCPTR_DELY	real	MAX_RCPTR
RCPTR_DELZ	real	MAX_RCPTR
RCPTR_HGT_REF	char*12	MAX_TRAJ
RCPTR_ID	char*12	MAX_RCPTR
RCPTR_INFL_X_FRACT	real	1
RCPTR_INFL_Y_FRACT	real	1
RCPTR_INFL_Z_FRACT	real	1
RCPTR_INTERVAL	char*(LEN_DUR_STRING)	1
RCPTR_MULT_IND	real	MAX_RCPTR
RCPTR_MULT_GLOBAL	real	1
RCPTR_MULT_GROUP	real	MAX_TRAJ
RCPTR_SBIN	integer	MAX_RCPTR
RCPTR_TITLE	char*70	1
RCPTR_TRAJ	integer	MAX_RCPTR
TRAJ_DATE	char*LEN_TIME_STRING	MAX_TRAJ_TIMES
TRAJ_TIME	char*LEN_TIME_STRING	MAX_TRAJ_TIMES
TRAJ_X	real	MAX_TRAJ_TIMES, MAX_TRAJ
TRAJ_Y	real	MAX_TRAJ_TIMES, MAX_TRAJ

TRAJ_Z	real	MAX_TRAJ_TIMES, MAX_TRAJ

ADPIC_SOURCE:

AMBIENT_TEMP	real	MAXTIM,MAXSOURC
CENTER_HGT	real	MAXTIM,MAXSOURC
CHEM_BIO_DECOMP_TAU	real	MAXTIM, MAXSOURC
DECAY_MODE	char*16	MAXSOURC
EXIT_VEL	real	MAXSOURC
HALFLIFE	real	MAXSOURC
HEAT_EMISSION_RATE	real	MAXSOURC
INV_HGT_AT_STACK	real	MAXTIM,MAXSOURC

NAMELIST VARIABLES	***TYPE***	***DIMENSIONS***
PART_DIAM_MAX	real	MAXSOURC
PART_DIAM_MEDIAN	real	MAXSOURC
PART_DIAM_MIN	real	MAXSOURC
PART_DIAM_SGD	real	MAXSOURC
PART_DENSITY	real	MAXSOURC
RELEASE_START	char*(LEN_DUR_STRING)	MAXSOURC
RELEASE_STOP	char*(LEN_DUR_STRING)	MAXSOURC
SFC_DEP_HGT	real	1
SFC_DEP_VEL	real	MAXSOURC
SIGX	real	MAXSOURC
SIGY	real	MAXSOURC
SIGZ	real	MAXSOURC
SOURCE_RATE	real	MAXTIM,MAXSOURC
SOURCE_START_DATE	char*(LEN_DATE_STRING)	MAXTIM
SOURCE_START_TIME	char*(LEN_TIME_STRING)	MAXTIM
STACK_RADIUS	real	MAXSOURC
STACK_TEMP	real	MAXSOURC
TOTPART	integer	MAXSOURC
USE_PLUME_RISE	char*1	MAXSOURC
VERT_TEMP_GRAD	real	MAXTIM,MAXSOURC
XLOC	real	MAXSOURC
XNEG_CUTOFF	real	MAXSOURC
XPOS_CUTOFF	real	MAXSOURC
YLOC	real	MAXSOURC
YNEG_CUTOFF	real	MAXSOURC
YPOS_CUTOFF	real	MAXSOURC
ZNEG_CUTOFF	real	MAXSOURC
ZPOS_CUTOFF	real	MAXSOURC

SIGMA_THETA_DATA:

SIGMA_THETA	real	MAXTIM, MAXLEVEL_ALL
SIGMA_THETA_HGT	real	MAXTIM, MAXLEVEL_ALL

ADPIC_EXPL_CLD:

DIF_IN_EXPL_CLD	char*1	1

DTDZ1	real	1
DTDZ2	real	1
HEAMT	real	1
HEATDET	real	1
PLOT_EXPL_CLD	integer	1
PRINT_EXPL_CLD	char*1	1
PRINT_TYPE_EXPL_CLD	char*5	1
REL_HUM	real	1
SIG_OVER_R	real	1
SRC_EXPL_CLD	integer	MAXSOURC,EXPL_CLD_DIM
SRC_PRESS	real	1
TA0	real	1
TEMP_PROFILE	real	MAX_EXPL_CLD_PROF
TEMP_PROFILE_HGT	real	MAX_EXPL_CLD_PROF
NAMELIST VARIABLES	***TYPE***	***DIMENSIONS***
UAEXP	real	1
UAREF	real	1
UAZREF	real	1
VEL_COUPL	real	1
ZI	real	1
ZINIT	real	1

CONC_ANALYSIS_CONTROL:		
ANGLE_WIDTH_FACTOR	real	1
CENTERLINE_SOURCE	integer	1
CONC_RESTART_DATE	char*(LEN_DATE_STRING)	1
CONC_RESTART_TIME	char*(LEN_TIME_STRING)	1
DCON	real	MAXFBINS
DOSE_TO_FINAL_BIN	integer	MAXDBINS,MAXFBINS
FULL_SEARCH_FACTOR	real	1
INTEGRATION_STEP	char*(LEN_DUR_STRING)	MAXPLOTSERIES,MAXFBINS
INTEGRATION_TYPE	char*4	MAXFBINS
PLOT_END_DATE	char*(LEN_DATE_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_END_TIME	char*(LEN_TIME_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_START_DATE	char*(LEN_DATE_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_START_TIME	char*(LEN_TIME_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_STEP	char*(LEN_DUR_STRING)	MAXPLOTSERIES,MAXFBINS
PRINT_CONTOUR_CONC	char*1	1
PRINT_INPUT_CONC	char*1	1
RADIAL_STEP_FACTOR	real	1
SAMPLING_TO_DOSE_BIN	integer	MAXDBINS
SECTOR_ANGLE_MAX	real	1
SECTOR_ANGLE_MIN	real	1
PLCNT_GRAPHICS_PARAMS:		
BLOWUP_SOURCE	integer	1

CIRCLE_2000	char*1	1
CONTOUR_LEVELS	real	MAXCONTLEVELS,MAXFBINS
HALF_POWER_OF_10	char*1	1
INITIAL_BANNER	char*1	1
MAPSCALE	integer	2
MAP_SCALE_FLAG	char*1	MAXFBINS
MAX_CONTOURS	integer	MAXFBINS
MIN_CONTOUR_AREA	real	1
NEST_CONTOURS	char*1	1
NEST_TYPE	char*10	1
NUM_RECOMPUTED_CONTOURS	integer	1
OUTPUT_GRIDS	char*1	1
RECOMPUTE_USER_CONTOURS	char*1	1
SOURCE_OCTANT	char*2	1
XMAX_WINDOW	real	1
XMIN_WINDOW	rea	1

NAMELIST VARIABLES	***TYPE***	***DIMENSIONS***
YMAX_WINDOW	real	1
YMIN_WINDOW	real	1

PLCNT_LABELING_PARAMS:

DESCR_FORMAT	char*5	MAXFBINS
DESCR_HEADERA	char*32	MAXFBINS
DESCR_HEADERB	char*32	MAXFBINS
GENERATION_DATE	char*(LEN_DATE_STRING)	1
GENERATION_TIME	char*(LEN_TIME_STRING)	1
LEGEND_TITLE	char*32	MAXFBINS
LEGEND_TITLE_ALL	char*32	1
LEVEL_DESCRit	char*32	5*MAXFBINS
MATERIAL	char*22	MAXFBINS
PROBLEM_TITLE_PLCNT	char*70	1
PLOT_TITLE	char*32	MAXFBINS
REMARKSt	char*32	4*MAXFBINS
REMARKSt_ALL	char*32	4
SOURCE_LOCATION_NUM	integer	1
UNITS	char*10	MAXFBINS

TIMEHIS_PARAMS:

BACKGROUND_CALC	real	MAXFBINS
BACKGROUND_CALC_ALL	real	1
BACKGROUND_MEAS	real	MAXFBINS
BACKGROUND_MEAS_ALL	real	1
BIN_PLOT_MAX	real	MAXFBINS
BIN_PLOT_MAX_ALL	real	1
BIN_PLOT_MIN	real	MAXFBINS

BIN_PLOT_MIN_ALL	real	1
COMPARE_MEAS_CALC	char*1	1
FACTOR_R	real	MAXFACTORS
HIT_ANGLE	real	MAXHANGLES
MIN_CONC	real	MAXFBINS
MIN_CONC_ALL	real	1
PROBLEM_TITLE_TIMEHIS	char*70	1
SHOW_TIME_ORIGIN	char*1	1
TIMESTEPS_PER_PLOT	integer	1
XMAX_SAMPLES	real	1
XMIN_SAMPLES	real	1
YMAX_SAMPLES	real	1
YMIN_SAMPLES	real	1

Appendix F. ARAC Operational Modeling System

The MATHEW/ADPIC modeling system, as described in the previous sections of this document, is a general purpose diagnostic dispersion modeling capability. Various changes have been made over the period from 1992 to 1995 to allow the system be moved to variety of computing platforms with minimal difficulty (with the exception of graphics, which remain problematic in terms of portability). However, the primary application of this system is in the ARAC Emergency Response Operating System (AEROS), which runs on VAX/VMS. This use of the models has resulted in the evolution of various conventions and procedures for their use in ARAC. Some of these issues, which affect the choice of parameters in the namelist files, have been described in the appropriate sections of this user's guide. Other conventions related to the execution of the codes in the VMS environment are particular to ARAC and the set of computing platforms that form the system. As these conventions and procedures are not documented elsewhere, the purpose of this appendix is to record this information for users of the ARAC central system.

1. *Computing platforms*

As mentioned above, the AEROS central system runs exclusively on VAX/VMS platforms. As a result two DEC-6000s running VMS are used by the central system. However, the desire to have increased computing power devoted to operational ARAC capabilities led to the acquisition of two DEC-Alpha machines. Somewhat later, a decision was made to re-engineer the entire ARAC system to take advantage of improved hardware and software capabilities, move to the Unix operating system, as well as to develop and import new modeling capabilities. This very large task (referred to as ARAC III) was identified as the priority effort in the program. As a result, the migration of the central system to the Alphas was abandoned as not being cost-effective (while not nearly as costly as the ARAC III effort, the task was not trivial due to certain technical problems such as archaic graphics). However, to take advantage of the compute power of the Alphas, the most computationally intensive components of the central system, i.e., the MEDIC, MATHEW and ADPIC models, were migrated to the Alphas. Access these versions of the models is limited to using batch submissions. The extra computing power is applied to the use of larger grids than is possible in the operational setting on the VAX machines. Thus, large grids are run on the Alphas while smaller grids continue to be executed on the VAXs as described below.

2. *Grid dimensions*

Traditionally, the regional MATHEW/ADPIC models have been run using standard grid dimensions. This reflects, in part, the limitations of Fortran 77 and earlier versions of Fortran with respect to dynamic dimensioning. Until recently, the models were compiled with the MEDIC and MATHEW codes using IMAXMAT of 51, JMAXMAT of 51 and KMAXMAT of 15. Allowing for a 5 cell buffer from the lateral boundaries of the wind grid, the ADPIC model was dimensioned with using IMAXPIC of 41, JMAXPIC of 41 and KMAXPIC of 15. Note that these dimensions refer to the number of grid points, with the number of grid cells being one less than the number of grid points. These values were chosen originally to meet the constraints of the hardware available as ARAC

became operational in the late 1970s, i.e., the models were run on CDC 7600s and these dimensions allowed the array of Lagrangian multipliers in MATHEW to just fit in Small Core Memory (SCM) on this machine allowing for the portion of SCM usurped by the operating system. This was an important optimization at the time. When the ARAC models were ported to VAX/VMS in 1982, these dimensions were maintained, initially to simplify comparisons with the 7600 versions. These dimensions have been continued to be used over the years as representing a reasonable compromise between the need for resolution in producing a quality model calculation and the need for speed in meeting the requirements of ARAC's emergency response customers. In particular, the current VAX machines allow the models to be executed in 5-10 minutes for typical first response scenarios, which allows ARAC to produce an initial product in 15 minutes for supported sites.

In the mid-1980s, responses to events affecting large areas, such as Chernobyl, led to the development of a variant version of the MATHEW/ADPIC codes that covered hemispheric scales. This version of the codes, HMATHEW/HADPIC (along with HTOPOG, HMEDIC, HPLCNT), were driven by the GWC hemispheric grids, which were provided on a 47 by 51 grid. Thus, both HMATHEW and HADPIC were compiled with an IMAX of 47, a JMAX of 51 and a KMAX of 15. By the time of the model restructuring, which occurred during 1992-1993, the capabilities of the regional MEDIC code for handling gridded metdata had generalized to the point that it was no longer necessary to maintain a separate hemispheric version of the codes. In addition, the substantial difficulty in recompiling with new dimensions that existed in the pre-restructured models was eliminated. As a result, the hemispheric models were essentially recompiled versions of the regional models (that last non-trivial differences between the two code versions disappeared when the ARAC map projection system went into production in 1994). The choice of these two versions of the ARAC models was reflected in the command mnemonics typed by ARAC assessors to execute the codes, e.g., STATOP (STArt model TOPog) and STAHTO (STArt model Hemispheric TOpog). These mnemonic commands initiate DCL scripts (*.COM files) that set the appropriate environment for the model and then execute the model. Both the regional and hemispheric model executables were maintained in the directory DSK$ARAC_ROOT:[PRODUCTN.MODEL] and the logical name MODEL$EXE was assigned to this directory.

With the introduction of the alpha capability, the conventions for handling versions of the models with varying grid dimensions have be upgraded to support larger numbers such grids in a more flexible way. When a user creates a problem (CREPRO), they are prompted to select one of a number of grids via a DCL menu as follows:

Select ADPIC computational Grid

1. Regional 41x41x15
2. Regional 51x51x31 (ETEX)
3. Regional 81x81x31 (Alpha)
4. Hemispheric 47x51x15

5. Hemispheric 95x101x31 (Alpha)

Enter Selection :

These grids are described by their ADPIC grid dimensions (in terms of grid points). When the user selects a grid, three logical names are defined. GRID$S_SIZE_DESC contains a string describing the grid (e.g., “41x41x15”) and GRID$S_SIZE_NAME contains a string that is used to differentiate the grids in the DCL that supports model executions. These strings are:

Regional 41x41x15	“R41”
Regional 51x51x31	“R51”
Regional 81x81x31	“R81”
Hemispheric 47x51x15	“H51”
Hemispheric 95x101x31	“H101”

These NAME strings will be used in the remaining discussion to refer to the various versions of the models. Note that the R41 grid is the traditional default regional model and the H51 grid is the usual hemispheric model. The R51 grid was used to run the European Tracer Experiment (ETEX) model runs. In this grid, there are more vertical levels and the MATHEW and ADPIC grids are the same (51x51x31). The R81 and H101 grids have twice the number of grid cells along each of the three coordinate axes as the default regional and hemispheric grids and so have eight times as many cells. The third logical name set when the user selects the computational grid is GRID$F_SIZE, which is assigned the name of a file (<*run_directory*>SELECTED_GRIDSIZE.DAT) in the run directory that records the values of GRID$S_SIZE_DESC and GRID$S_SIZE_NAME, so that the choice of grid is remembered whenever that particular run is reselected. [Aside: note that this same mechanism is used to remember the *.GRID file associated with a run in the file <*run_directory*>SELECTED_GRID.DAT. In addition, the map family, map member and elevation file associated with a problem are remembered in the files SELECTED_MAPFAMILY.DAT, SELECTED_MAPMEMBER.DAT and SELECTED_TOPO.DAT, respectively, which are maintained in the problem journal directory, i.e., DSK$ARAC_SITE_ROOT:<node_id>.<problem_id>.JOURN. The map projection is remembered by the presence of the map projection description file in the JOURN directory for a problem]. The models for each of these computational grids reside in a sub-directory of the MODEL$EXE directory with the sub-directory having the name of the particular grid, e.g., the R81 grids are maintained in the directory DSK$ARAC_ROOT:[PRODUCTN.MODEL.R81]. This directory is referred to with the logical name MODEL$EXE_SUBDIR in the various DCL scripts that execute the models.

3. *Large grid modeling on the VMS-ALPHAs*

As mentioned above, the two large grids, R81 and H101, are executed on the VMS-Alphas using the batch queues. At one level, this does not affect the user interaction with the models; STAMED,

STAMAT and STAADP are typed by the user and the scripts determine how to run the model on the basis of the logical name GRID$S_SIZE_NAME. However, there are some differences that affect how the user determines the status of a model execution and investigates problems that occur in such an execution. These differences are discussed in this section. Note that for the R41, R51 and H51 grids, the procedure has not changed, with the code executing on the VAX, the output to the terminal appearing on the screen and the observed metdata typically being read from the Problem Station Library and the Problem Metdata Files.

As mentioned previously, when the user has selected the R81 or H101 grids, a batch process is submitted to the appropriate queue on a VMS-Alpha that initializes the necessary logical names to create an appropriate environment on the Alpha and executes the code on that machine. One implication of this is that the normal messages sent by the model to the screen do not appear. These messages are written to the log file associated with the batch submission. This log file appears in the run directory (note that the VMS Alphas share the same disks with the VAX machines and thus can read and write files in the run directory) and is named BATCH_ALPHA_<*model*>_<*arac_user_name*>.LOG, e.g., a MEDIC run on an Alpha under DCL in BETAHOYT would generate a log file with the following name: BATCH_ALPHA_MEDIC_HOYT1.LOG. The use of arac_user_name in the filename is intended to avoid confusion in the synchronization of model runs, as described below, when several users are running models on the Alpha.

While same disks and files are seen from both the VAXs and Alphas, global sections on the VAX are not accessible on the Alphas. Thus, the Problem Station Library is not available for use on a VMS-Alpha. To circumvent this difficulty, STAMED automatically runs a program (Build_ASCII_Probmet, which is available to the user directly using the CREASM mnemonic) that builds a STNLOC.MET and an OBSERV.MET file from the Problem Station Library and Problem Metdata Files on the basis of the MEDIC.NML file in the run directory if the MEDIC.NML file indicates that the Problem Station Library is to be used, i.e., STNLOCPSL = 'Y' in MEDIC_CONTROL. Note that, at present, if STNLOCPSL = 'N' then it is assumed that an OBSERV.MET file already exists, thus the STNLOCPSL flag controls the creation of both the STNLOC.MET and the OBSERV.MET files (this will eventually be upgraded to allow the extraction of a STNLOC.MET file or an OBSERV.MET file independently). Also note that MEDIC.NML should always contain appropriate METDATASET strings for each MEDIC_PARAMS. The MEDIC model parameters system provides these strings. These requirements on the MEDIC.NML allow the same namelist file to be used for both the Vax and Alpha versions of the code. Relying on differentially compiled versions of the code, the Alpha MEDIC always looks for a STNLOC.MET and an OBSERV.MET file, i.e., STNLOCPSL is ignored. The Alpha version also ignores any PROBMET_DST information. When MEDIC is run on a Vax, these variables have their normal effect and the METDATASET strings are ignored because the VAX MEDIC checks the PROBMET_DST variables before checking the METDATASET variable and relies on the Problem Metdata Files only if the PROBMET_DST variables are set. As long as STNLOCPSL is set to 'Y',

new STNLOC.MET and OBSERV.MET files are created each time STAMED is executed with a large grid so that any editing of the MEDIC.NML file with respect to metdata is reflected in the run performed.

When a batch process is submitted, there is no direct way to determine when the model completes. Thus, a mechanism is needed to synchronize the running of an Alpha model with the Vax DCL script that initiates the model execution. In addition, the values of the various logical names used by the models must be passed to the command file when it begins executing on the Alpha. While DCL allows parameters to be passed to a batch process in the SUBMIT command, the number of parameters is limited to eight, which is not sufficient to set the model run environment. The two-way communication needed to control the model run is provided via simple text files. The arguments needed to initialize the model environment are written to a file on the VAX from the DCL script that runs a model, e.g., START_MODEL-MEDIC.COM. This file is placed in the run directory and is contains the following information:

1. Problem_id
2. Node_id
3. Run_id
4. Model_subdir
5. MODEL$A_GRID
6. MODEL$F_GRID
7. MODEL$A_WINDS
8. MODEL$A_DISPERSION
9. MODEL$A_VLST
10. DECAFGWC$A_AFGWC_GRID_FILE
11. DSK$MEDB_ROOT
12. DSK$ARAC_RWM_MET
13. DECNODDS$A_NODDS_GRID_FILE

This parameter file is called ALPHA_<*model*>_<*arac_user_name*>.PARAMS. The SUBMIT parameters are used to pass the name of the parameter file to the batch process as well as the necessary information to create the synchronization file in the correct location, i.e., the run directory, the arac user name and the model name. Note that the use of a file to pass all the parameters allows additional information to easily be passed to the batch process if it necessary to support new modeling capabilities. Synchronization between the submitting script on the Vax and the batch process is achieved by having the Vax script attempt to read a status file every five seconds until it finds such a file. This file is written by the batch process on the Alpha on completion of the run and

reports success or failure. The file is called ALPHA_<*model*>_<*arac_user_name*>.STATUS and is written in the run directory (note that at the beginning of the Vax DCL script, all previous versions of the matching status file are deleted to avoid confusing the synchronization), e.g., the status of an Alpha MEDIC execution will be in the file ALPHA_MEDIC_HOYT1.STATUS and will contain the string "MEDICSUCCESS" if MEDIC completed correctly and the string "MEDICFAILURE" if MEDIC failed. Note that the use of this file naming convention means that different arac users can be working in the same run directory without stepping on each other's parameter and status files.

So in summary, the evaluation of the performance of an Alpha model is made be the examination of the appropriate status, parameter and log files (note that the log file includes some basic system performance statistics). There is no way to detect whether a submitted large grid model is actually running directly on a Vax. It may often be helpful to bring up a separate window and log into the appropriate Alpha and run MONITOR PROCESS/TOPCPU from DCL to check that the batch run is actually executing (note that the batch process name is listed on the screen on the Vax as the process is submitted and that the name of the log file is also listed on the screen on the Vax).

4. *Graphics*

Graphics on the Vax are created by each model execution with the graphic products written to a graphics metafile (CGM) file. These CGM files are processed with a program (Product_View, run with the mnemonic SHOGRAMOD) that displays the graphics and allows them to be routed to a PostScript printer. This program relies on a SmartStar interface. The GKS graphics package on which this capability is archaic and could not be cost-effectively ported to the Alphas. As a result, an alternative graphics strategy is used in which PV-WAVE scripts are used to access and display the contents of the *.BIF files that are used to transfer information between models. These PV-WAVE graphics only apply to the large grid MEDIC, MATHEW and ADPIC model. TOPOG, PLCNT and TIMEHIS run on the Vax machines so their graphics is unaltered except for the greater resolution. Also note that the PARTPOS files produced by ADPIC behave normally and are displayed as usual with PERSPEC (SHOTER) on the Vax.

Index (Input Files)